

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Sistemas de Controlo e Supervisão de Sistemas Embebidos – Tipo SCADA

Ivo António Ferreira

Mestrado Integrado em Engenharia Electrotécnica e de
Computadores

Porto, Março de 2008

Sistemas de Controlo e Supervisão de Sistemas Embebidos – Tipo SCADA

Ivo António Ferreira

Dissertação realizada no âmbito do Mestrado em Engenharia
Electrotécnica e de Computadores da Faculdade de Engenharia da
Universidade do Porto, orientado pelo Professor Doutor Adriano da Silva
Carvalho e o Engenheiro Celso Cavadas

(O Presidente do Júri: Professor Doutor Américo Lopes de Azevedo)

Faculdade de Engenharia da Universidade do Porto

Departamento de Engenharia Electrotécnica e de Computadores

Março de 2008

Agradecimentos

Gostaria de agradecer a todos os que me ajudaram durante a realização desta dissertação, não só pela ajuda mas também pela preocupação demonstrada.

Resumo

Actualmente, a obtenção de sucesso no mundo empresarial é fortemente influenciada pela integração plena de todos os constituintes do sistema produtivo, de forma a permitir uma gestão realmente eficaz de todo o sistema. Nesta necessidade de integração, os sistemas de SCADA desempenham um papel de elevada relevância, ao permitirem de forma simples e rápida, a obtenção de dados relativos aos sistemas de produção, possibilitando assim um maior e melhor controlo sobre este, que se reflectirá no aumento da sua eficácia e qualidade. Apesar das enormes potencialidades existentes, neste tipo de sistema, a sua implementação continua a ser relativamente dispendiosa.

Com este trabalho, pretende-se avaliar a possibilidade de implementação deste tipo de sistemas recorrendo apenas a um simples computador e ao uso do protocolo USB, que devido ao seu uso generalizado poderá permitir a implementação de soluções ao nível económico mais competitivas e também mais flexíveis, sendo esta avaliação, validada com a implementação de uma plataforma que possibilite o suporte de sistemas deste género.

Assim, começar-se-á por fazer um enquadramento das soluções existentes actualmente, para a implementação deste tipo de sistemas, sendo também realizada uma descrição geral das tecnologias utilizadas para permitir a integração do todo o sistema produtivo. Na parte introdutória deste trabalho, fazer-se-á também uma descrição geral das características fundamentais do protocolo USB.

Na fase seguinte, será feita uma análise de requisitos da plataforma implementada, para que fosse possível a validação deste estudo. Sendo também aqui, enunciadas e justificadas as escolhas de projecto que conduziram esta implementação. Seguidamente, será feita uma descrição da implementação desta plataforma, sendo relatados pormenores destas ao nível do hardware e software desenvolvidos, sendo também apresentados e analisados alguns resultados obtidos.

Por último, será feita a apresentação das conclusões obtidas neste estudo, sendo também apresentados alguns passos futuros que poderão melhorar a plataforma desenvolvida.

Abstract

Currently, the achieving of success in the business world is heavily influenced by the total integration of all components of the productive system in order to allow a truly effective management of the whole system. In this need of integration the SCADA systems play a role of high relevance. They permit a faster and simpler way to obtain data regarding to the production system, allowing a better control over the process. This will be reflected in an increase of efficiency and quality. Despite the huge potential in this type of system, its implementation is still relatively expensive.

This work is intended to assess the possibility of implementing this kind of systems using a PC and USB protocol. Due to the generalized use of the USB protocol new low-priced, more competitive and flexible solutions will arise. This assessment may be validated by the implementation of platform supporting these systems.

It will be described some of the existing solutions to implement these kinds of systems. Also it will be described the technologies that allow integrating all the components of the productive system. In the introductory part of this work, make will be also a general description of the main features of the USB protocol.

In the next stage, it will be done an analysis of the requirements of the platform implemented, so that is possible to validate this work. Being here too, listed and justified the choices of project that led to implementation. Lately, it will be done a description of the implementation of the platform containing detailed description of the hardware and software developed. Some the results obtained it will be analyzed and discussed.

Finally, will be presented the conclusions achieved with this work. It will be indicated some possible changes in order to improve the platform in the future.

Índice

Agradecimentos	3
Resumo	4
Abstract.....	5
Índice	6
Índice de Figuras	9
Índice de Tabelas	10
Capítulo 1 Introdução	11
1.1 Motivação	11
1.2 Contextualização.....	11
1.3 Objectivos.....	12
1.4 Estrutura da Dissertação	13
Capítulo 2 Estado da arte.....	14
2.1 Sistemas de Controlo	14
2.2 Tecnologias de Controlo: Evolução	16
2.3 Redes Industriais	19
2.3.1 Redes de Campo	21
2.4 SCADA.....	24
2.4.1 Descrição Funcional	27
Capítulo 3 USB.....	30
3.1 Protocolo de Comunicação	32

3.2 Configuração do Sistema.....	33
3.3 Gestão do Acesso ao Barramento.....	34
3.4 Tipo de Tráfego	35
3.4.1 Tráfego de Controlo.....	36
3.4.2 Tráfego Isócrono.....	37
3.4.3 Tráfego de Interrupções.....	38
3.4.4 Tráfego <i>Bulk</i>	39
Capítulo 4 Análise de Requisitos.....	40
4.1 Protocolo de interligação dos dispositivos de Campo.....	40
4.1.1 Protocolo I ² C	41
4.1.2 Protocolo SPI.....	44
4.1.3 Protocolo CAN	47
4.1.4 Escolha do Protocolo a utilizar.....	48
4.2 Escolha do microcontrolador.....	49
4.3 Escolha dos periféricos.....	50
4.4 Resumo da solução Proposta	51
Capítulo 5 Implementação.....	53
5.1 Circuito Produzido.....	53
5.1.1 Entradas e Saídas digitais	57
5.1.2 Entradas Analógicas	58
5.1.3 Expansão.....	59
5.2 Microcontrolador	60
5.3 Protocolo de Comunicação Série.....	63

5.4 Protocolo Comunicação SPI.....	67
5.4.1 Comunicação com as extensões de portas.....	67
5.4.2 Interação entre os dois microcontroladores	71
5.5 Interação com o LCD.....	73
5.5 Aplicação do Computador	75
5.5.1 Plataforma de Desenvolvimento.....	76
5.5.2 Funcionamento	77
5.6 Resultados.....	80
5.6.1 Velocidade de Leitura das Portas Digitais.....	80
5.6.2 Resposta a Comandos vindos do Computador	81
5.6.3 Conclusões.....	83
Capítulo 6 Conclusões	84
Referências	87

Índice de Figuras

Figura 1- Arquitecturas de controlo	15
Figura 2- Stack das redes de campo	22
Figura 3- Topologia Física USB.....	31
Figura 4- Troca de dados SPI	45
Figura 5- SPI topologia “ <i>Daisy Chain</i> ”	46
Figura 6- SPI topologia comum	46
Figura 7- Arquitectura Funcional do Sistema.....	52
Figura 8- Circuito aquisição de dados	55
Figura 9- Circuito de controlo do LCD	56
Figura 10- Entradas e Saídas Digitais.....	58
Figura 11- Entradas analógicas.....	59
Figura 12 - Fluxograma do funcionamento do microcontrolador	61
Figura 13-Trama de dados série	63
Figura 14-Máquina de estados série	66
Figura 15-Byte de comando comunicação Periféricos	68
Figura 16-Tabelas que reflectem a organização dos registos [6]	69
Figura 17-Trama SPI comunicação entre dois micros.....	71
Figura 18- Formato do Comandos para escrever no LCD.....	74
Figura 19-Máquina de estados da comunicação entre os dois micros para controlar LCD visto do emissor.....	75
Figura 20- Funcionamento em modo manual.....	78
Figura 21-Funcionamento em modo automático.....	79
Figura 22-Imagem de medição do tempo de leitura das portas digitais	81
Figura 23-Variação do estado de uma saída com o envio de comandos de escrita de 100 em 100ms.....	82
Figura 24-Variação da saída com comandos de 10 em 10ms.....	83

Índice de Tabelas

Tabela 1- Descrição da função das tramas porta série.....	65
Tabela 2-Tipo de comandos usado na comunicação entre os dois microcontroladores .	72

Capítulo 1 Introdução

1.1 Motivação

Este trabalho foi feito no âmbito do estágio curricular desenvolvido na HIDROMAC. O objectivo inicial deste trabalho era o desenvolvimento de um protótipo para uma plataforma de testes de dispositivos electrónicos, baseada num PC, que se pretendia que futuramente substitua a plataforma existente. Com a plataforma a desenvolver, desejava-se tornar o sistema mais rápido e mais flexível, surgindo por isso a ligação ao computador utilizando o protocolo de comunicações USB, que permitia comunicações mais rápidas que o método actual, e conferia maior mobilidade e flexibilidade ao sistema, visto este tornar-se totalmente externo ao computador.

1.2 Contextualização

Os sistemas de SCADA actualmente têm um grande domínio de aplicação, desde aplicação no âmbito industrial, percorrendo um vasto conjunto de indústrias, até, por exemplo, aplicação em equipamentos de serviço público, como sistemas de tratamento de água, sistemas de transmissão de energia ou sistemas de transportes, passando também por aplicações na área da domótica, como por exemplo, sistemas de ar condicionado ou sistemas de alarme.

A sigla SCADA é um acrónimo para expressão “*Supervisory Control and Data Acquisition*”. Este tipo de sistemas recolhe dados de vários sensores, espalhados ao longo do processo em questão, e centraliza-os num computador central, que os gere e apresenta aos operadores do sistema. Com isto, permite a criação de “fotografias” do estado do processo, permitindo assim, de forma simples, uma monitorização contínua deste. Isto assume importância bastante relevante, por exemplo, em processos, que devido à sua dimensão espacial ou devido a dificuldades de acessibilidade, a sua monitorização de outra forma era muito difícil ou bastante dispendiosa. Por outro lado, esta função é bastante relevante em situações de funcionamento anómalo do processo, pois assim de forma rápida se consegue detectar essa situação e mais rapidamente a

corrigir, e assim melhorar o desempenho do sistema. Outra funcionalidade importante, é a possibilidade, de através da utilização de uma base de dados, com um sistema de SCADA, se conseguir guardar um histórico do funcionamento do processo monitorizado, permitindo assim obter estatísticas sobre o seu comportamento, que poderão ser utilizadas para melhorar o desempenho deste, de forma que se torne mais produtivo e rentável. Justificando-se assim a sua larga utilização hoje em dia.

Apesar das grandes vantagens apresentadas pelos sistemas de SCADA, estes, hoje em dia, continuam a ser sistemas relativamente caros, pois, normalmente, estes são implementados recorrendo a PLCs (*Programmable Logic Controller*) e a computadores industriais, que apesar de cumprirem de forma bastante eficaz as suas funções, ainda são algo dispendiosos e criam soluções pouco mutáveis.

Tendo em conta o que foi dito até aqui, neste trabalho pretende-se estudar e desenvolver uma plataforma, que desempenhe funções de supervisão, do tipo SCADA, recorrendo a um PC e ao protocolo de comunicação USB, com o objectivo de minimizar os contras apontados a este tipo de sistemas hoje em dia.

O facto da escolha do protocolo de comunicação a utilizar, ter recaído sobre o USB, deve-se às velocidades de transmissão que este pode atingir, para a versão USB2.0 máximo de 480Mb/s, e ao facto do seu uso ser generalizado.

1.3 Objectivos

Os objectivos deste trabalho consistem no estudo da viabilidade da implementação e uso de sistemas de supervisão e controlo, do tipo SCADA, suportados pelo uso do protocolo de comunicação USB e de um simples PC. Para validação experimental deste estudo pretende-se, projectar e implementar uma plataforma que permita suportar um sistema do tipo SCADA, utilizando as tecnologias referidas anteriormente. Esta plataforma tem, como requisitos mínimos, permitir a monitorização e controlo, de pelo menos, 16 entradas digitais, 16 saídas digitais e 8 entradas analógicas. Para interligação dos diferentes dispositivos de campo, pretende-se, pelo menos, que se considere o uso do protocolo I2C, não sendo, obrigatório o seu uso caso se conclua que existem melhores soluções. Pretende-se também que esta a plataforma, implementada permita possíveis expansões do sistema.

1.4 Estrutura da Dissertação

No capítulo 2 desta dissertação, será feito um ponto de situação sobre o estado da arte relativamente às redes industriais, com enfoco nas redes de campo, e aos sistemas de SCADA, sendo dada uma perspectiva da sua evolução ao longo do tempo.

No 3º capítulo, é feito uma descrição sobre o funcionamento do USB. Essa descrição incide em aspectos como por exemplo, o funcionamento do protocolo que regula a transmissão de dados neste barramento, ou no tipo de tráfegos cuja transmissão é suportada. É feita também a descrição, de como é feita a gestão do acesso ao barramento.

No 4º capítulo, é feito o levantamento dos requisitos da plataforma que se pretende desenvolver. É feito também a justificação das diferentes opções de projecto tomadas e enunciados os dispositivos escolhidos.

No capítulo 5, é descrita a implementação da plataforma desenvolvida e os resultados obtidos. São descritos todos os aspectos relevantes da implementação, com enfoco especial para a implementação da comunicação entre os diferentes dispositivos utilizados. Será focado também, o funcionamento dos dispositivos onde se centra a “inteligência”, o microcontrolador e a aplicação desenvolvida para o computador.

Por último, no 6º capítulo, são enunciadas e discutidas as conclusões obtidas e apresentados alguns dos possíveis melhoramentos que podem ser feitos à solução final obtida.

Capítulo 2 Estado da arte

2.1 Sistemas de Controlo

Os sistemas de controlo e observação são essenciais nos processos industriais actualmente. Estes são responsáveis pelo seu funcionamento de forma correcta. Este tipo de sistemas pode ser dividido em 3 constituintes essenciais: o processo a controlar, o controlador e o operador humano [1].

O controlador assume uma elevada importância no sistema de controlo, pois é este que é responsável pela execução de, forma correcta, do algoritmo de controlo mais adequado ao funcionamento correcto do processo a controlar. Para cumprir a sua função da melhor forma possível, o controlador necessita de interagir com as outras duas partes do sistema de controlo, o processo que se pretende controlar e operador. A interacção com o processo tem como objectivo cumprir a função final do sistema, o controlo do seu funcionamento. Esta interacção é feita através de um conjunto de sinais que permitem ao controlador, por um lado obter informações sobre o estado do sistema, vindos de sensores, de forma a poder decidir o que fazer para garantir o funcionamento correcto do processo, por outro, permiti-lhe por em prática as decisões tomadas, através da manipulação de actuadores ligados ao processo.

Por outro lado, a interacção com o operador humano, pretende permitir que este tenha acesso ao estado do processo e hipóteses de, caso seja necessário, executar algumas funções de controlo, como por exemplo, configurar alguns parâmetros do sistema, de forma a otimizar o seu funcionamento, ou poder intervir em casos de funcionamento anómalo. Esta interacção é conseguida recorrendo a uma interface Homem-Máquina, normalmente constituídas por um mecanismo de visualização do estado do sistema e de mecanismos de actuação sobre o sistema.

A nível industrial existe uma enorme variedade de tipos de processos a controlar, cada um deles com características específicas, que obrigam o controlador tenha que ser projectado e implementado de maneira a cumprir a sua função da forma mais eficaz possível. Um dos aspectos mais importantes deste tipo de projecto centra-se na

arquitectura funcional que suporta o funcionamento do controlador. Podem ser definidas 3 arquitecturas de controlo diferentes:

- Arquitectura Centralizada, onde o algoritmo de controlo é executado por um único dispositivo;
- Arquitectura Descentralizada – Esta arquitectura caracteriza sistema onde o algoritmo de controlo é executado por um único dispositivo, mas em que algumas funções de aquisição de dados são distribuídas por dispositivos menos complexos, implicando redes de transmissão de dados normalmente ponto-a-ponto.
- Arquitectura Distribuída – Aqui a execução do algoritmo de controlo é distribuída por diferentes dispositivos, com a mesma complexidade ou não. Este tipo de estrutura implica também a existência de redes de comunicação, mas neste caso, redes de maior complexidade, como as redes de campo [1].

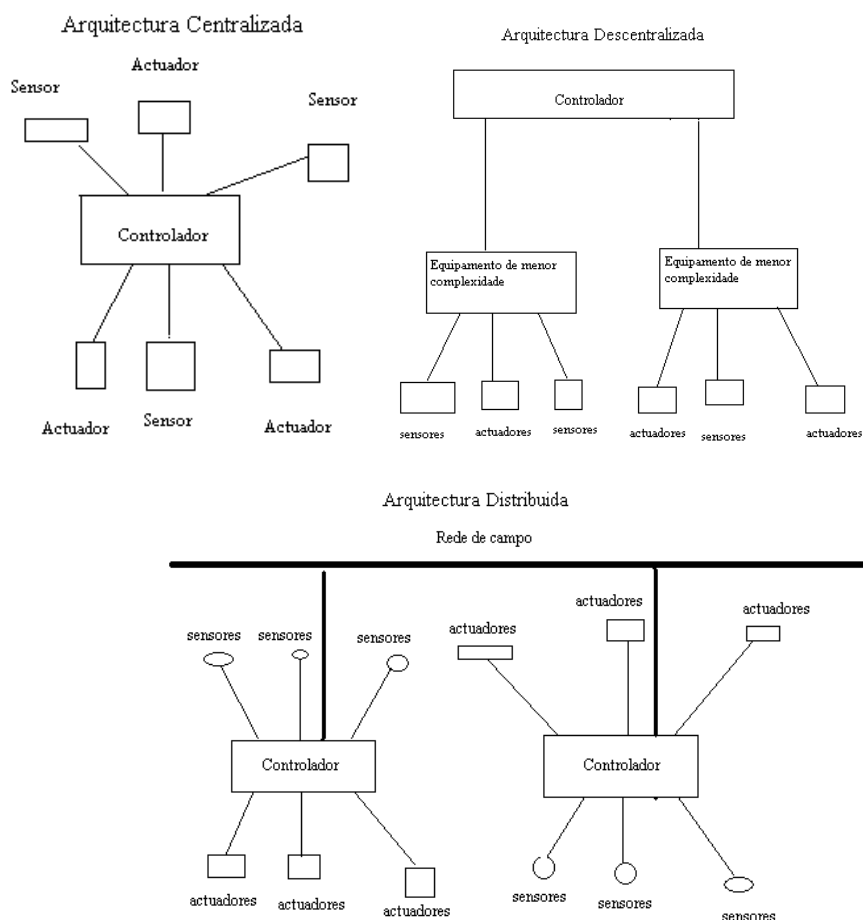


Figura 1- Arquitecturas de controlo

2.2 Tecnologias de Controlo: Evolução

Os primeiros sistemas de controlo surgiram nos primórdios da civilização. Estes eram muito rudimentares sendo utilizados nos sistemas mecânicos utilizados na altura para executar algumas funções, por exemplo na agricultura.

Com o surgimento da revolução industrial, os sistemas utilizados sofreram uma grande evolução. Conceitos, como produção massiva e o surgimento das linhas de produção implicaram o início da automatização deste tipo de sistemas, utilizados, nesta altura, principalmente em funções pouco complexas e repetitivas.

O surgimento da energia eléctrica, no final do século XIX, impulsionou a um novo desenvolvimento dos sistemas de controlo. Este desenvolvimento manifestou-se com a progressiva substituição dos sistemas mecânicos por relés eléctricos. Esta nova tecnologia permitiu a criação de sistemas mais eficazes e complexos, permitindo também a sua aplicação a sistemas cada vez mais complexos.

Entre as décadas de 40 e 60 surgiram dois novos acontecimentos que permitiram uma grande evolução no projecto e implementação dos sistemas de controlo. Estes foram o surgimento tanto do transístor, como dos circuitos impressos. Estes levaram ao surgimento da lógica discreta. Estas novas tecnologias levaram à criação de sistemas mais eficientes e mais rápidos, assim como à miniaturização destes. Por outro lado, apesar do aumento da sua eficiência e eficácia, estes sistemas, ao assentarem na ligação de dispositivos eléctricos e electrónicos, apresentavam grandes limitações ao nível da sua adaptabilidade e capacidade de evolução, pois ao assentarem na ligação física dos diferentes constituintes, qualquer mudança implicava alterações nestas, o que se tornava muito custoso e complicado.

O surgimento do computador, nos anos 60, trouxe novamente uma grande evolução aos sistemas de controlo. Com este, passou a ser possível a centralização num único dispositivo todas as funções de um sistema de controlo, desde a aquisição de dados até à interface como os operadores. Por outro lado, o facto de estes serem programáveis, permitiu que se comesasse a produzir sistemas genéricos, que com trabalho ao nível da programação, poderiam ser aplicados a diferentes sistemas, que se

pretendessem controlar. Também por esta altura, apareceram os primeiros sistemas de controlo numérico e os primeiros robots industriais [1].

Apesar das vantagens introduzidas pelo uso do computador, até este momento as arquitecturas de controlo assumiam uma estrutura centralizada. Com o aparecimento do microcontrolador, passou a ser possível transferir algumas das funcionalidades dos computadores para estes novos dispositivos, possibilitando alguma descentralização dos sistemas de controlo. Estas novas possibilidades evoluíram segundo duas perspectivas: na indústria de processos, foram utilizadas para o desenvolvimento de sistemas de controlo distribuído (*DCS- Distributed Control Systems*), que permitissem a interligação de dispositivos de controlo mais simples, como PIDs, aos de maior complexidade de forma hierárquica [1]; por outro lado, na indústria da manufactura, foram utilizadas para que os PLCs se tornassem no elemento central dos sistemas de controlo [1].

Apesar dos grandes desenvolvimentos sentidos nos diferentes constituintes dos sistemas de controlo, havia um aspecto que se manteve praticamente inalterado, as comunicações entre os diferentes dispositivos. Estas baseavam na utilização de comunicações série analógicas, como os anéis de corrente, e comunicações digitais, como ligações 0-24V. Este tipo de comunicações, além de não permitir grandes taxas de transmissão e permitir apenas a transmissão de pequenas quantidades de informação, baseava-se essencialmente em ligações ponto-a-ponto, o que implicava a utilização de grandes quantidades de cablagem, o que dificultava em muito o projecto e manutenção destas redes.

Com a grande evolução sentida nas redes industriais, este problema foi solucionado. Esta evolução foi sentida, de forma mais intensa, ao nível da interligação dos dispositivos de campo. O desenvolvimento das redes de campo possibilitou a substituição das ligações ponto-a-ponto, utilizadas até então, pela implementação de barramentos, que possibilitavam a interligação de diferentes dispositivos. Esta substituição acabou com a utilização de uma grande quantidade de cabos para permitir a interligação de todos os dispositivos, introduzindo assim grandes melhorias ao nível do projecto, implementação e manutenção destas redes de comunicações. Por outro lado, o uso das redes de campo permitiu a introdução de melhorias significativas ao nível da performance, maiores taxas de transmissão e maiores quantidades de informação, entre outras. Esta melhoria na performance contribuiu de forma decisiva para a

implementação de sistemas de controlo cada vez mais eficazes. A utilização deste novo método de comunicação contribuiu também, para a descentralização cada vez maior das arquitecturas de controlo utilizadas.

A não existência de padronização nesta área das redes industriais levou a uma proliferação de soluções de implementação deste tipo de redes, muitas delas proprietárias. Isto fez com que muitas vezes fosse impossível ou muito complicado a interligação de dispositivos de diferentes fabricantes, limitando assim as potencialidades destas na interligação de dispositivos. Este problema foi resolvido apenas no decorrer desta década, com o surgimento de dois standards, o IEC 61158 e o IEC 61784, onde são contemplados as soluções mais importantes existentes no mercado.

Por outro lado, no final dos anos 90, iniciou-se uma introdução no ambiente industrial, de redes de comunicações, que até então, apenas se utilizavam em ambiente não industrial. Esta introdução deveu-se, entre outros, ao facto de estas soluções possuírem interfaces de comunicação mais baratas e melhor desempenho que as soluções existentes, além de que, se baseavam em protocolos standards [1]. O exemplo mais flagrante é a Ethernet, com a sua versão industrial.

O desenvolvimento das redes de comunicações industriais, não só ao nível da interligação dos dispositivos de campo, permitiu o surgimento de novos conceitos, como o CIM (*Computer Integrated Manufacturing*), que previam a interligação de todos os sectores de uma empresa, de forma a permitir uma gestão verdadeiramente integrada do sistema produtivo.

Apesar do surgimento de conceitos como o CIM e o desenvolvimento das redes industriais, a integração vertical de um sistema de produção ainda não era um problema debelado. Os diferentes requisitos de comunicação das diferentes funções existentes neste processo, impede de se utilizar uma solução única que consiga cobrir todas elas, implicando, que para a obtenção deste objectivo, sejam desenvolvidas soluções de integração entre as diferentes redes, sendo que o seu desenvolvimento exige um grande esforço tanto ao nível do hardware como do software. Para tentar facilitar esta integração, foram desenvolvidos modelos por camadas, baseados no modelo OSI, como por exemplo o modelo MAP e o MSS, que nunca tiveram grande sucesso devido à falta de suporte tecnológico. Na indústria de processos, este problema foi tentado resolver através da utilização de sistemas de SCADA.

O aumento de necessidade de integração com os sistemas de gestão de produção e planeamento, no final dos anos 90, fez com que surgissem novas ferramentas de software que possibilitam a criação de soluções bastante válidas em termos de integração, horizontal e vertical. Estas soluções baseiam-se numa arquitectura de objectos distribuídos, sendo os exemplos mais populares o CORBA (*Common Object Request Broker Architecture*), solução com vários domínios de aplicação e o OPC (*OLE for Process Control*), mais focalizado para ambientes industriais [1], existindo nesta altura uma tentativa de criação de standards para este tipo de tecnologia, o IEC 61499 e o IEC 68804 [1][2].

Neste momento, a investigação centra-se na utilização das tecnologias *Web* nos sistemas de controlo, como por exemplo HTTP, Java, XML, UDP/TCP/IP, entre outros, de modo a conseguir uma integração completa nas camadas de topo destes sistemas e melhorar a sua integração com aplicações externas a estes [1].

2.3 Redes Industriais

As redes industriais actualmente assumem um papel fundamental no mundo empresarial. Através destas, é possível integrar todas as áreas constituintes de uma empresa de forma simples, rápida e eficaz, contribuindo assim de forma bastante importante para a melhoria da qualidade, eficácia e lucros de uma empresa, através da redução dos custos de produção, distribuição e venda dos produtos.

Apesar de as redes industriais se estenderem cada vez mais por todas as áreas de uma empresa, os requisitos de comunicação e os fluxos de informação em cada uma delas são muito distintos, impedindo a adopção de uma solução única que cobrisse todas elas. Assim, a solução procurada é a selecção das redes a utilizar em cada uma das áreas de forma a cumprirem os requisitos locais de cada uma das áreas da empresa e que o sistema global consiga cumprir com eficácia a função da integração da empresa como um todo.

Como as arquitecturas de controlo, todas as actividades ligadas ao controlo do processo industrial assumem uma estrutura hierárquica. Assim as comunicações industriais, como têm como objectivo final a obtenção da optimização do controlo do

processo industrial, assumem uma estrutura hierárquica, intimamente ligada à estrutura de controlo dos processos. Este modelo hierárquico é caracterizado por fluxos de informação dentro do mesmo nível, como também fluxos entre níveis adjacentes [1]. Assim tendo em conta a estrutura hierárquica das comunicações industriais, podemos dividi-las em 3 tipos diferentes: redes de fábrica, redes de célula e redes de campo.

As redes de fábrica designam as redes utilizadas para suprimir as necessidades de comunicação das áreas de gestão e de engenharia numa empresa, fornecendo serviços de comunicação com as restantes áreas da empresa mas também com o exterior a estas, como serviços de acesso à internet ou e-mail. Tendo em conta o modelo hierárquico das comunicações industriais, as comunicações com os níveis inferiores, no sentido descendente, baseiam-se na comunicação de ordens de fabrico e do seu escalonamento, enquanto no sentido oposto, centram-se em dados relativos ao comportamento do processo de fabrico, como por exemplo, o estado de ordens de fabrico, ou dados relativos ao desempenho deste, que permitam análises relativas à sua qualidade e que possam levar ao seu melhoramento [1]. Os requisitos necessários para este tipo de rede centram-se essencialmente na capacidade de transmitir grandes quantidades de informação, que não têm grandes requisitos ao nível temporal. Do ponto de vista do utilizador, apesar de não existirem grandes requisitos temporais, pretende-se que esta rede que seja relativamente rápida, evitando grandes esperas no envio de informações ou em acessos na internet, o que leva a que seja necessário estas redes suportarem elevadas taxas de transmissão, actualmente na ordem dos Gbit/s.

Por sua vez, as redes de célula situam-se no ponto central da hierarquia das redes industriais, tendo como função principal fazer a interligação entre os dois níveis a si adjacentes. Estas são responsáveis por traduzir as ordens de fabrico, vindas das redes de fábrica, para ordens de execução de determinadas tarefas a serem enviadas aos dispositivos situados no nível inferior, assim como, pela compilação dos dados recolhidos do nível inferior, relativos à evolução das tarefas ordenadas e ao desempenho do sistema, para depois os enviar para o nível superior. Este nível é normalmente constituído por equipamentos de média complexidade como autómatos, sistemas de SCADA, robots, entre outros. Estas redes devem suportar a transmissão de quantidades de informação mediana, com requisitos temporais não muito rígidos.

Por último temos as redes de campo. Estas são utilizadas para interligar os dispositivos que se encontram junto do sistema de produção e que são responsáveis pelo seu funcionamento e controlo. Assim estas redes são usadas para interligar autómatos, sensores, actuadores, sistemas embebidos, entre outros. Devem suportar requisitos temporais muito rígidos, tempo real, essencial para que os sistemas de controlo a este nível funcionem devidamente, transmissão de pequenas quantidades de informação mas muito frequente.

2.3.1 Redes de Campo

Antes do surgimento das redes de campo, as ligações entre os dispositivos de campo era feita utilizando o tipo de ligações standards existentes na altura como os anéis de corrente 4-20mA e as ligações 0-24 V. Este tipo de ligação implicava o uso de um grande número de cablagem, o que dificultava o seu projecto, implementação e manutenção, além de ser um método bastante dispendioso. Com o surgimento das redes de campo pretendia-se solucionar esta situação, através da criação de um barramento que permitisse a ligação de todos os dispositivos, que levasse à diminuição da cablagem usada e que diminuísse também os custos de projecto, de implementação e de manutenção das redes de comunicação a este nível. Pretendia-se também, criar um sistema de comunicações mais eficaz e que permitisse mais facilmente modificar e alterar a sua estrutura, permitindo facilmente a interligação de novos dispositivos.

Como foi dito antes, as redes de campo são utilizadas para interligar os dispositivos mais próximos do sistema de produção, assumindo um papel muito importante na implementação dos sistemas de controlo ao nível da produção. Devido à sua função, estas necessitam de ser implementadas em ambiente industrial. Este tipo de ambiente é bastante hostil para as comunicações. Trata-se de um ambiente ao nível mecânico e electromagnético muito exigente, com elevado ruído electromagnético, devido ao elevado número de máquinas, e com condições ambientais muito desfavoráveis, com a existência de grande sujidade, temperaturas elevadas, grandes percentagens de humidade, entre outros. Existem também ambientes, que devido ao tipo de matéria-prima e produtos fabricados, se tornam bastante exigentes ao nível da segurança do próprio ambiente e das pessoas que o frequentam, como por exemplo, a indústria

petroquímica, onde o perigo de explosão está sempre presente. Assim, além dos requisitos impostos pela função de suportar o controlo dos processos industriais, a sua implementação no ambiente industrial impõe requisitos em termos mecânicos e electromagnéticos às redes de campo. Por outro lado, na indústria o objectivo principal é a obtenção de lucro, exigindo-se que os meios utilizados para a implementação do processo industrial sejam fiáveis, e que em caso de possíveis avarias, estas sejam rapidamente resolvidas e que afectem da menor forma possível todo o processo. Então, os principais requisitos das redes de campo são:

- Capacidade de transferir pequenas capacidades de informação, mas transmitidas frequentemente, de forma eficiente [1];
- Suportar tráfego periódico, assim como tráfego aperiódico (eventos), cumprindo os requisitos temporais, tempo real, exigidos por estes;
- Imunidade a ruídos electromagnéticos e protecção mecânica contra condições ambientais muito hostis, como temperaturas elevadas, poeiras, humidade, etc.
- Elevados níveis de fiabilidade, disponibilidade e segurança;
- Baixo custo global por nó;

Baseando-se no modelo OSI mas tendo em conta os requisitos das redes de campo, foi criada uma modelo que pudesse representar os diferentes aspectos relacionados com este tipo de redes. Assim, foi criada uma estrutura de 3 camadas, sendo estas: a camada de aplicação, a camada de ligação de dados e a camada física.



Figura 2- Stack das redes de campo

Apesar de criado este modelo, não foram definidos nenhuns standards de como seriam cumpridos os requisitos e funções de cada uma das camadas. Por isso, foram

desenvolvidas muitas soluções de implementação das funções de cada uma das camadas, tendo em conta essencialmente o tipo de aplicação para onde se orientavam cada uma das soluções desenvolvidas. Isto levou, a que houvesse uma proliferação de diferentes soluções.

A falta de standards, que levou à proliferação de numerosas soluções, maioritariamente proprietárias, fez com que se tornasse quase impossível a integração de diferentes soluções e a utilização de dispositivos de diferentes fabricantes. Para resolver esta situação, no início dos anos 90, iniciou-se um processo de criação de uma solução standard. Este processo foi bastante demorado, durou mais de 10 anos, devido a uma tentativa dos diferentes intervenientes de criar um standard baseado ou muito próximo da sua solução. Este processo concluiu-se com o estabelecimento de dois standards internacionais: o IEC 61158 e o IEC 61784, cada um deles baseados no estabelecimento de perfis de comunicação, que contemplam as mais importantes na altura [1] [2].

Neste momento, resolvido o problema da criação de um standard para as redes de campo, devido ao aumento do número de dispositivos usados em qualquer solução, os fabricantes apostam na criação de novas ferramentas que permitam facilitar o desenvolvimento e configuração de novas redes. Também nesta área, existem problemas ao nível da padronização das soluções de desenvolvimento e configuração. Existindo maioritariamente soluções proprietárias, que não permitem a integração de diferentes soluções. Este problema espera-se que seja resolvido com o uso do conceito de *function blocks*, conceito já standard [3].

2.4 SCADA

SCADA é o acrónimo para a expressão “*Supervisory Control And Data Acquisition*”. Este tipo de sistemas permite monitorizar e controlar vários tipos de sistemas. Isto é feito através da recolha de dados relativos ao sistema em causa, que depois são compilados, tratados e apresentados ao utilizador, permitindo-lhe também a possibilidade de exercer algumas funções de controlo sobre este.

Nos anos 20, o desenvolvimento da meteorologia e da indústria aeronáutica tornou necessário a recolha de grandes quantidades de dados, para posterior análise, em locais de difícil acesso, por exemplo, em aviões e foguetões, cujos voos iniciais normalmente acabavam em despenhamentos, ou locais inóspitos, onde colocar seres humanos era extremamente difícil e perigoso. O problema que surgiu nessa altura era como seria feita a comunicação desses dados para uma estação central, onde estes poderiam ser analisados e tratados. A solução para este problema, foi obtida através do sistema utilizado no sistema ferroviário para controlar a aproximação de comboios a bifurcações e controlar o caminho que estes deveriam tomar. Estes sistemas utilizavam um sistema de comunicações por fio, que através da colocação de *switchs* eléctricos junto das linhas, que ligados a uma estação central conseguiam manipular o estado de indicadores luminosos, indicando se existia algum comboio em aproximação e o estado da bifurcação, permitindo aos controladores do tráfego dar ordens se seria ou não necessária alterar o caminho escolhido na bifurcação. Este sistema poderá ser considerado o primeiro sistema de telemetria, sistemas que deram origem aos sistemas de SCADA.

A solução utilizada no sistema de controlo de tráfego rodoviário, era uma excelente solução para sistemas fixos e cuja mobilidade era nula. Mas o problema mantinha-se para sistemas onde a estação de recolha de dados era móvel. Por esta altura, dava-se também uma significativa evolução na tecnologia rádio, que se apresentava como uma possível solução para este problema.

Com o passar do tempo, a tecnologia rádio foi evoluindo tornando-se mais fiável e permitindo a transmissão de maiores quantidades de dados. Mas a tecnologia rádio continuava a ter uma grande limitação, o facto de permitir apenas transmissão de dados

unidireccional. Este facto era bastante limitativo, pois apenas permitia a recolha de dados [4].

Ao mesmo tempo, as comunicações baseadas em fio foram evoluindo. Essa evolução conduziu a uma melhoria da fiabilidade deste tipo de comunicações, mas mais importante ainda, foi a introdução do conceito de comunicações bidireccionais. Esta evolução revelou-se de extrema importância, ao permitir, não só, a recolha de dados remotamente mas também a possibilidade de efectuar algumas funções de controlo, também remotamente. O próximo passo nos sistemas de telemetria centrou-se na aplicação da comunicação bidireccional à tecnologia rádio, aumentando assim o seu alcance.

Os sistemas de telemetria sofreram um grande empurrão no seu desenvolvimento com o aparecimento dos computadores digitais, nos anos 60. Esta nova tecnologia permitiu uma maior flexibilidade a este tipo de sistemas, principalmente ao nível da estação central, onde eram centralizados os dados. Antes do aparecimento dos computadores e para suportar o desenvolvimento dos sistemas de comunicação, as estações centrais dos sistemas de telemetria tinham se tornado extremamente complexas e grandes [4]. Com uso dos computadores digitais, foi possível centralizar todos os dados recolhidos no terreno, tornando estes sistemas mais simples e eficazes.

No início dos anos 70, foi usada pela primeira vez a designação de SCADA [4]. Nesta década, deu-se também uma grande evolução na tecnologia rádio o que levou a que fosse substituindo cada vez mais as comunicações com fios.

Também durante esta década, houve uma grande evolução principalmente ao nível do software o que resultou em melhores interfaces com o operador, começando a aparecer os primeiros sistemas de produção de relatórios de funcionamento dos sistemas, o que permitia obter toda a informação pretendida sobre o sistema.

À medida que os computadores se foram tornando mais potentes, a evolução dos sistemas de SCADA foi-se acentuando cada vez mais. Além disso, com a redução do preço dos computadores, usados essencialmente nas estações centrais dos SCADAs, estes deixaram de necessitar de ser centralizados, havendo agora a hipótese se necessário de se distribuir o controlo dos sistemas [4].

Este desenvolvimento levou a uma proliferação destes sistemas, fazendo com que a obtenção de frequências para suportar as comunicações dos sistemas via rádio, se torna-se cada vez mais difícil, passando a haver uma aposta em outras tecnologias de comunicação sem fios, como as comunicações via satélite [4].

A partir desta altura até aos dias de hoje, o desenvolvimento dos computadores e dos sistemas de comunicação permitiram que fossem aumentando as funcionalidades dos SCADAs. Ao nível dos sistemas de comunicação, o aparecimento das LANs (Local Area Network), onde as velocidades de transmissão aumentaram consideravelmente, permitiu que a interligação dos sistemas de SCADA com as ferramentas de gestão de produção e financeira das empresas melhorasse. Com estas redes, tornou-se possível também a interligação a sistemas de base de dados. Isto fez com que os registos históricos, já existentes, se tornassem verdadeiramente eficaz, e se tornassem ferramentas poderosas na gestão das empresas e nos sistemas de gestão de qualidade. Além disso, estas redes permitiram distribuir e espalhar as funções de visualização e de controlo associadas às estações centrais, permitindo o acesso simultâneo a estas funcionalidades a várias pessoas dentro das empresas. Associado ao uso destas redes, o desenvolvimento das tecnologias ligadas à *Web* aumentou ainda mais o alcance dos SCADAs, fazendo com que hoje seja possível, a que um gestor de uma determinada empresa, tenha acesso às informações disponibilizadas pelo sistema, em qualquer dia e qualquer hora e qualquer lugar, através do uso da Internet. Por outro lado, estas novas possibilidades levantaram novos problemas ao nível da segurança. Estes problemas derivam do facto de, com o uso da Internet, os sistemas de SCADA terem sido abertos ao exterior da empresa, o que implica que seja necessário montar bons sistemas de segurança nas redes internas das empresas de modo a impossibilitar que pessoas exteriores à empresa e não autorizadas possam aceder a informações importantes, que possam por em risco o funcionamento da empresa.

Ao nível das comunicações, o desenvolvimento das redes industriais e das redes de campo, em particular, permitiu melhorar as comunicações ao nível das estações remotas, tornando assim mais eficaz a recolha dos dados e a execução das funções de controlo necessárias. O desenvolvimento destas tecnologias melhorou também a interligação das diferentes estações remotas e a estação central.

A evolução sentida ao nível dos computadores, levou a uma melhoria significativa das interfaces, de visualização e controlo com o operador. Estas melhorias sentiram-se essencialmente no aspecto visual destas, tornando-se mais apelativas e amigáveis para o utilizador, e na maior capacidade de manipulação dos dados recolhidos, permitindo fornecer ao utilizador outros tipos de informação, como informação estatística, e outras formas de visualização da mesma, como por exemplo gráficos.

2.4.1 Descrição Funcional

Como já foi enunciado, os sistemas de SCADA têm como função fornecer a possibilidade ao utilizador de poder visualizar o estado de um determinado processo e executar funções de controlo sobre este. O controlo permitido por estes sistemas é um controlo supervisionado. Este tipo de controlo consiste na execução de acções de controlo, sobre malhas de controlo que funcionam de uma forma independente, com o objectivo de otimizar o seu funcionamento. Exemplos de este tipo de acções são o estabelecimento de novas referências de funcionamento. Para executar estas funções, estes sistemas têm 3 constituintes: as RTU (*Remote Terminal Unit*), a MTU (*Master Terminal Unit*) e os interfaces homem-máquina (HMI).

As RTUs são as unidades que se encontram junto do sistema que se pretende monitorizar e controlar, podendo ser apenas uma ou em sistemas bastante complexos atingirem a ordem das centenas. Estas são responsáveis pela interacção com os sensores e actuadores utilizados. Considerando o conceito de controlo supervisionado, é nestas estações que são implementados os algoritmos de controlo locais existentes nos sistemas de SCADA. Para a implementação destas estações normalmente recorre-se ao uso de PLCs, mas podem ser usados também microcontroladores para desempenhar este papel. Além de interagirem com os sensores e actuadores, estão também ligadas à MTU, a quem vão fornecer o estado das variáveis que estão a controlar. Este facto implica que estas guardem em memória o valor de todas as variáveis, de forma que a unidade central quando o desejar possa acede-las. A qualidade de cada uma das RTUs de um SCADA é avaliada essencialmente pela fiabilidade, segurança e qualidade dos dados que transmite.

A MTU (*Master Terminal Unit*) é o centro do sistema de SCADA. Esta é responsável pela interface entre as RTUs e os interfaces Homem-Máquina, funcionando muitas vezes como uma gateway de comunicações. Ela é responsável pela leitura dos dados recolhidos pelas RTUs e pela tradução destes para o contexto do sistema, de forma a poderem ser utilizados pelos interfaces HMI, para fornecer a “imagem” do estado do sistema. São responsáveis também pela comunicação em sentido inverso, interpretando os comandos enviados pelo operador e comunicá-los à RTU, para que esta possa actuar da forma desejada sobre os actuadores. Actualmente, outra funcionalidade importante da MTU, é permitir a ligação a uma base de dados, onde possa guardar a informação recolhida ao longo do tempo, de forma que possa ser utilizada pelos HMI, para disponibilizar informação sobre a evolução histórica do sistema, dados estatísticos, entre outros, ao operador. Com esta funcionalidade, os sistemas de SCADA tornaram-se uma ferramenta bastante interessante, para ser usada em sistemas de gestão de produção e de gestão de qualidade, pois permitem uma análise completa do sistema em questão.

Por último, temos as interfaces Homem-Máquina. Estas são o interface do sistema de SCADA com o exterior. Estas têm como função, apresentar aos operadores uma “imagem” do sistema. Normalmente, esta “imagem” é apresentada de forma gráfica, através de esquemas que representam os diferentes constituintes do sistema e apresentam informações sobre o seu estado de funcionamento, como por exemplo, um esquema que represente uma bomba de água e apresente o fluxo que está a ser bombeado em tempo real. Normalmente, as HMI permitem ao utilizador não só visualizar o estado do sistema com também manipular determinadas variáveis deste, permitindo assim o controlo supervisionado. Na apresentação do estado dos sistemas, os HMIs devem tratar de forma especial as situações de alarme. Estas situações devem ser indicadas o mais depressa possível e de forma bem visível, de forma a o sistema de SCADA poder desempenhar da melhor maneira possível umas das suas principais funcionalidades, o facto de permitir de forma rápida a detecção de situações anómalas, de forma a estas poderem ser resolvidas o mais rápido possível aumentando a rentabilidade e eficácia do processo que está a ser controlado. Outra das funcionalidades, que as interfaces com os operadores devem suportar é a possibilidade dos operadores poderem analisar o comportamento do sistema ao longo do tempo, através de um registo histórico e de dados estatísticos, que possam ser usados para melhorar o desempenho do sistema. Outro dos pontos que deve ser levado em

consideração, no projecto de interfaces Homem-Máquina, é a segurança. Esta é uma questão importante, para evitar que pessoas não autorizadas, executem acções para as quais não tenham competência e ponham em causa o desempenho do sistema. Esta questão maior relevância pelo facto de, hoje em dia, através das redes locais, existirem em simultâneo vários interfaces a serem acedidos por diferentes operadores, que normalmente acedem a zonas do processo diferentes, aonde devem ser usados sistemas de segurança, com o uso de *passwords* que limitem o acesso dos diferentes utilizadores às acções e dados que realmente podem aceder.

Capítulo 3 USB

O USB (*Universal Serial Bus*), como o próprio nome indica é um barramento série, que permite a conexão de diferentes dispositivos entre si. Este foi desenvolvido para criar uma interface standard, que permitisse que diferentes tipos de dispositivos se ligassem a um computador através da mesma interface, sem que fosse necessária a instalação de drivers específicos para cada um dos dispositivos. Outro dos objectivos do desenvolvimento do USB, era a criação de uma interface que suportasse conexões *Plug and Play*, de forma a ser extremamente fácil ligar e desligar dispositivos do barramento. Pretendia-se também que fosse possível através do barramento alimentar dispositivos de baixo consumo que se conectassem a este, de modo a que estes não precisassem de uma fonte de alimentação externa.

O barramento USB é constituído por 4 linhas. Duas linhas de alimentação, VBus, 5V, e massa, e duas linhas de transmissão de dados D+ e D-, sendo a transmissão destes feita em modo diferencial, estando as duas linhas entrelaçadas de modo a reduzir o ruído electromagnético, funcionando a comunicação em modo *half-duplex*. Neste momento, existem 3 versões diferentes deste protocolo, sendo elas, a versão 1.0 (*Low Speed*), versão 1.1 (*Full Speed*) e versão 2.0 (*High Speed*). As velocidades de transmissão máximas permitidas por estas são respectivamente, 1.5Mbit/s, 12Mbit/s e 480Mbit/s.

No barramento USB, existem 3 tipos de dispositivos: o *Host*, os periféricos e a Hub. O *Host* está no computador e é este que controla o funcionamento de todo o barramento. Este inicia todas as transmissões de dados e controla as conexões dos dispositivos no barramento e sua configuração, entre outras funções. Os periféricos representam todos os componentes conectados ao barramento. Na versão *on-the-go* (OTG) é possível os periféricos optarem se pretendem assumir o papel de periférico ou de *host*, possibilitando assim a ligação de dois periféricos sem recorrer ao uso de um computador.

Os periféricos têm como objectivo fornecer uma determinada funcionalidade extra ao *host*. As funcionalidades fornecidas podem ser bastantes diversas, podendo ir desde as funcionalidades fornecidas por um rato até às fornecidas por uma impressora.

Apesar disso, em termos lógicos, a interface com os *host* é idêntica para todos os periféricos, permitindo assim que este possa tratar todos os periféricos de igual forma independentemente do seu tipo. Além disso, com o intuito de facilitar o funcionamento do *host*, cada periférico contém informações necessárias para que este o possa identificar e configurar. Existe também a hipótese de um dispositivo poder fornecer mais do que uma função ao *host*.

O último constituinte do sistema de USB é a Hub. A Hub é um dispositivo cuja função é fornecer ao *host* pontos de conexão extras, permitindo assim a multiplicação destes. Devido à funcionalidade fornecida por si, estas desempenham um papel muito importante na topologia física estabelecida no sistema USB, como será descrito posteriormente.

Em termos físicos, o sistema USB assume uma topologia de estrela por camadas. Por constrangimentos temporais, o número de camadas possível está limitada a um máximo de 7, sendo apenas possível conexões que recorram a 5 *hubs* que não se encontrem no *host*. Por outro lado, o número de periféricos que poderão estar conectados ao barramento é limitado aos 127. Esta topologia foi definida de forma a impedir uma ligação circular, que não é permitida devido ao funcionamento do barramento. Como foi dito anteriormente, esta topologia é apenas possível devido à existência hubs, que fornecem pontos de ligação extras ao *host*, sendo que este pode ter também embebido uma hub, a *root* hub, que é o que acontece em computadores com mais que um ponto de ligação.

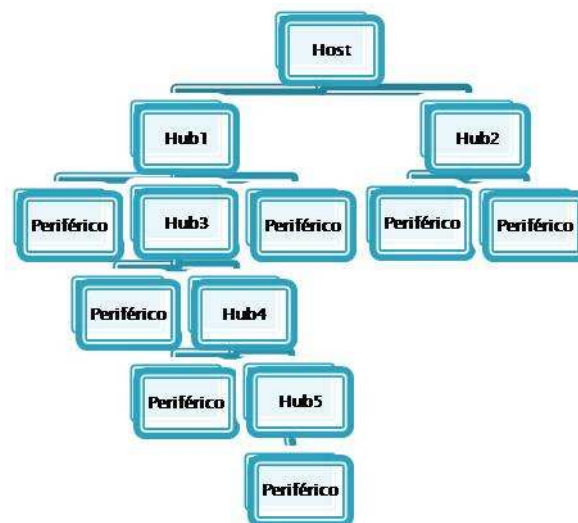


Figura 3- Topologia Física USB

Na versão 2 do USB, devido ao suporte de periféricos que funcionem com versões anteriores, os hubs desempenham um papel ainda mais importante, em termos físicos, pois permite a criação de braços que funcionem a velocidades diferentes dentro do mesmo barramento, sem ser necessário a instalação no *host* de controladores extras que suportem as diferentes velocidades [5].

Em termos lógicos, o protocolo USB funciona como se todos os dispositivos estivessem ligados ao *host* directamente, através da criação de “túneis” lógicos para a transmissão dos dados. Do ponto de vista lógico do barramento, um periférico é constituído por um conjunto de *endpoints*, que representam cada uma das funcionalidades fornecidas e que são os responsáveis pelo estabelecimento das ligações do lado do periférico.

3.1 Protocolo de Comunicação

No barramento USB, todas as comunicações são iniciadas pelo *host*. A grande maioria das trocas de dados é constituída pelo envio de 3 pacotes entre o *host* e o periférico. As trocas de dados são iniciadas pelo envio por parte do *host* de um pacote com a indicação do endereço do dispositivo e o número do *endpoint* com o qual pretende comunicar, sendo que este pacote se define como o “*token packet*”. Neste pacote, é indicado também qual é a direcção em que a transmissão de dados irá ocorrer, ou seja, se será do *host* para o periférico ou do periférico para o *host*. Seguidamente, ao envio do “*token packet*”, dá-se a transmissão da trama de dados pretendida. Caso não exista no emissor dados a ser transmitidos, este deverá transmitir uma trama indicando esse facto. Em qualquer um dos casos, o receptor responde com uma trama de *handshaking* de forma a confirmar a recepção da trama enviada. A troca destes 3 pacotes entre o *host* e o *endpoint* designa-se por transacção.

A transmissão dos dados em USB, entre o *host* e um determinado periférico, é feita através dos “túneis” lógicos referidos na descrição da topologia lógica do barramento. Existem dois tipos de “túneis”: *stream* e os de mensagem. A diferença entre os dois tipos de “túneis” é o facto de, nos de mensagem os dados serem transmitidos usando uma estrutura definida pelo protocolo, enquanto nos do tipo *stream* não. Além desta classificação, as ligações lógicas existentes no USB podem ser caracterizadas

também pelo tipo de transferências que suportam, pela largura de banda a si atribuída e pelas características dos *endpoints*.

3.2 Configuração do Sistema

Como foi dito anteriormente, o barramento USB suporta mudanças na sua topologia física a qualquer altura, por isso o software do *host* que controla a configuração deste, tem de estar preparado para o reconfigurar. Nas secções seguintes serão descritos os processos de conexão e desconexão de dispositivos do barramento, assim como o processo de configuração destes.

Como referido anteriormente, as conexões de novos dispositivos é feita através das portas de um hub. Estes têm bits que indicam o estado de cada porta que os constituem. O *host*, ciclicamente, pede para que os hubs lhes indiquem o estado desses bits. No caso de ter havido uma conexão, o *host* irá activar a porta e endereçar o dispositivo através de uma ligação de controlo utilizando o endereço pré-definido, o endereço 0. Seguidamente, irá atribuir um endereço ao novo periférico e determinar se este é um hub ou um outro tipo de dispositivo. No caso de este ser um hub e estarem outros dispositivos conectados às suas portas, o processo descrito repetir-se-á, se não for um hub, o processo de conexão será tratado pelo software do *host* mais apropriado para este tipo de função.

No caso de um dispositivo ser desconectado de uma das portas de um hub, este irá desactivar a respectiva porta, e informar o *host* do sucedido. Ao receber esta informação, a desconexão será tratada pelo software próprio do *host* para esta função. No caso do dispositivo desconectado ser um hub, o *host* terá de tratar da desconexão de todos os dispositivos a este conectados [5].

O processo de configuração do sistema quando há conexões e desconexões de dispositivos é conduzido pela actividade *Bus Enumeration*. Seguidamente será descrito todo este processo.

Primeiramente a hub onde é conectado o dispositivo informa o *host* que houve uma mudança de estado numa das portas, que lhe irá inquirir de que tipo de evento se trata, ou de uma conexão ou uma desconexão, e em que porta é que este ocorreu. Neste momento, o *host* sabe em que porta do hub está conectada o novo periférico. O próximo

passo, é activar esta porta e mandar um comando de *reset* para este. Antes desta acção, o *host* irá esperar pelo menos, 100ms, de modo a que o processo de inserção termine e a alimentação do novo dispositivo estabilize [5].

Depois de executado o comando de *reset*, o dispositivo irá responder pelo endereço pré-definido, o 0. A próxima acção será atribuir-lhe o endereço definitivo. Este processo ocorre depois de o *host* pedir-lhes informações sobre a quantidade de dados máxima que a ligação *default* pode usar.

Por fim, o *host* vai ler as informações de configuração do dispositivo e configura-lo de maneira própria, atribuindo a largura de banda necessária à ligação deste dispositivo e permitindo que este consuma toda a corrente necessária, entre outras, ficando assim o dispositivo pronto a ser utilizado.

3.3 Gestão do Acesso ao Barramento

Como qualquer transferência de dados, no USB, é sempre iniciada pelo *host*, é também ele que gere como é feito o acesso ao meio. Este processo implica a interacção de diferentes entidades do lado do computador, sendo elas: o software cliente, o driver USB e driver controlador do *host*.

O software cliente representa as diferentes aplicações a correr no computador, que necessitam de aceder aos diferentes periféricos que se encontram conectados ao barramento USB, sendo estas que decidem quando necessitam que ocorra uma transferência de dados. Quando isto acontece, a aplicação envia um pedido ao driver USB para que tal aconteça. Isto é feito pelo envio de um IRP (*I/O Request Packet*). Neste pacote, a aplicação indica todas as informações relevantes para que a transmissão possa ocorrer, entre elas, o endereço do dispositivo, o número do *endpoint*, o tamanho dos dados a transmitir, onde se encontram ou onde vão ser guardados os dados e o sentido da transmissão. No processo de envio de tramas, o driver USB não tem um papel muito activo limitando-se a reencaminhar este para o driver do controlador do *host* (HCD).

Por sua vez, o HCD ao receber um IRP vai coloca-lo na lista de transacções. Nesta lista são colocadas informações sobre todas as transacções que é necessário realizar. Depois de colocar os IRPs na lista de transacções, o HCD vai seguir o seu

progresso de modo a garantir que os requisitos de largura de banda e do tempo das *microframes* são cumpridos. Ao verificar que o IRP foi concluído com sucesso o HCD irá notificar o software cliente desse facto.

A última entidade envolvida neste processo é o controlador do *host* (HC). Este vai ser responsável por fazer com que os IRPs inseridos na lista de transacções possam transformar-se realmente em transmissões de dados. Além de controlar o envio dos dados pelo barramento, garante que são cumpridas todas as regras de acesso ao meio. Por outro lado, este fornece informações sobre o estado de cada um dos IRPs, que depois serão usadas pelo HCD, para notificar a aplicação que o pedido de transmissão que fizera já fora atendido.

Como foi dito anteriormente, o controlador do *host* é responsável por transformar os IRPs na transmissão efectivas dos dados, sendo responsável pela sua organização em pacotes. Por vezes, o atendimento a um IRP implica que sejam executadas mais que uma transacção. Além da organização dos dados em pacotes, o HC é também responsável por gerir o escalonamento das diferentes transacções no barramento USB. Nesta gestão, apesar de um IRP poder ser constituído por várias transacções, o HC não é obrigado a efectua-las de forma consecutiva, podendo gerir a execução das diferentes transacções, dos diferentes IRPs, livremente, tendo apenas que cumprir os constrangimentos relativos ao funcionamento do protocolo de comunicação e relativos a cada um dos tipos de tráfego.

3.4 Tipo de Tráfego

O barramento USB suporta a transferência de diferentes tipos de tráfego. Neste barramento, uma transferência de dados consiste no transporte dos dados entre um buffer situado no *host* para outro associado ao *endpoint* respectivo. Este dados, como já referido, são transmitidos por “túneis”, sendo que dependendo do tipo de ligação utilizada, podem ser transportados em estruturas definidas pelo sistema USB ou não. Em qualquer um dos casos, estes são “empacotados” de forma previamente definida.

Como já foi dito atrás, o USB permite diferentes tipos de tráfego, que têm diferentes características e cumprem determinados requisitos, de forma a poder suportar

os mais diversos tipos de serviço. Os tipos de tráfego podem ser caracterizados pelos seguintes aspectos:

- Formato dos dados;
- Direcção da comunicação;
- Limites do tamanho dos pacotes;
- Constrangimentos de acesso ao barramento;
- Constrangimentos de latência;
- Sequência de dados requerida;
- Tratamento de erros [5];

No USB estão definidos 4 tipos de tráfego:

- Tráfego de Controlo;
- Tráfego Isócrono;
- Tráfego de Interrupções;
- Tráfego *Bulk*;

Nas secções seguintes, serão descritas as características principais de cada um dos tipos de tráfego.

3.4.1 Tráfego de Controlo

Este tipo de tráfego foi concebido para suportar comunicações relacionadas com a gestão do barramento entre o *host* e a função. As comunicações deste tipo de tráfego são feitas usando apenas uma transacção. Na primeira trama da transacção, o *host* envia para o periférico uma trama indicando o tipo de comando que pretende que seja executado. Seguidamente, é trocada uma trama de dados cuja direcção é definida na primeira trama enviada pelo *host*. Por fim, o periférico envia uma trama para o *host*, no final de ter executado o comando pedido, indicando que este foi executado com sucesso.

Este tipo de tráfego, é feito através de “túneis” do tipo mensagem, ou seja, as tramas usam uma estrutura de dados pré-definida no protocolo USB. Todos os dispositivos, ao serem conectados ao barramento, estabelecem uma ligação para troca de tráfego de controlo, sendo que se necessitarem podem criar mais ligações. Estas ligações suportam tráfego bidireccional.

Para a transmissão do tráfego de controlo, nem o *host* nem o periférico, podem reservar largura de banda ou determinar uma frequência para acesso ao barramento. Assim o sistema USB utiliza uma estratégia de “*best effort*” para transmitir estes dados [5]. Apesar de o periférico não poder especificar nada em relação ao acesso ao meio neste tipo de tráfego, o USB reserva uma percentagem de tempo por microframe para ser utilizado pelo tráfego de controlo, sendo essa percentagem de 20%. Além do mais, caso o número de transacções deste tipo pendentes seja superior à percentagem de tempo reservada para efeitos da sua transmissão se existir largura de banda disponível no resto do tempo, esta pode ser aproveitada para transmitir tráfego de controlo.

3.4.2 Tráfego Isócrono

O tráfego isócrono tem como objectivo suportar aplicações que necessitem de funcionar em tempo real, como por exemplo, transmissão de som e imagem. Assim neste tipo de tráfego, as questões temporais são de grande importância, por isso mesmo aqui a criação dos dados, o seu envio e a sua recepção são feitos de forma contínua e em tempo real, sendo que a sua transmissão é feita a uma taxa fixa. Outro aspecto importante, são os possíveis atrasos induzidos pela transmissão deste tipo de tráfego, por isso, em caso de falha de entrega de alguma trama não é utilizado qualquer mecanismo de retransmissão de dados, pois aqui torna-se mais importante a entrega dos dados sem atrasos mas possivelmente com erros, do que receber toda a informação correcta mas sem cumprir os requisitos temporais necessários [5]. Assim tendo em conta isto, este tipo de tráfego garante:

- Acesso à largura de banda com latência limitada;
- Velocidade de transmissão constante enquanto existirem dados a transmitir;
- Em caso de falha, não existem tentativas de retransmissão da trama cujo envio não foi feito com sucesso [5];

A transmissão deste tipo de dados não utiliza uma estrutura de dados definida, utilizando assim ligações do tipo *stream*. As ligações usadas são sempre unidireccionais, o que implica a criação de duas ligações para casos onde a transferência de dados seja tanto do periférico para o *host* como também em sentido contrário.

Os *endpoints* que utilizam este tipo de tráfego definem o número de dados que podem transmitir em cada trama. Este valor está relacionado com a sua capacidade de armazenamento, sendo que estão limitadas a um valor de 1024 bytes para o USB2.0. Por outro lado, os *endpoints* normais podem apenas enviar uma trama por *microframe*, a mais pequena divisão temporal no sistema USB para o USB2.0, tendo como valor 125µs. Existe um tipo especial de *endpoints*, os *high-bandwidth endpoints*, aos quais é permitido enviar 3 tramas por *microframe*, ou seja, até 3072 bytes.

Além do tamanho máximo de dados, que podem transmitir por trama os *endpoints* definem também o período máximo de acesso ao meio esse período é definido pela expressão $(2^{bInterval-1}) * F$, onde *bInterval*, é uma variável cujo valor é definido pelo *endpoint* e pode assumir um valor de 1 até 16 e F é o valor de uma *microframe*, 125µs. Tendo em conta isto, um *endpoint* que suporte tráfego isócrono pode transmitir até 3072 bytes por *microframe*, ou seja, transmitir a uma taxa máxima de 192Mb/s [5].

3.4.3 Tráfego de Interrupções

Este tipo de tráfego foi concebido para ser usado por aplicações cujos dados a transmitir ou receber não se produzem de forma cíclica, mas que quando surgem devem ser comunicados num espaço de tempo nunca superior ao definido pelo *endpoint*. Casos típicos de dados deste tipo são a comunicação de eventos, usado por exemplo, por ratos de computadores. Assim, um dispositivo que use tráfego deste tipo vê garantido:

- O período de serviço do pedido tem um valor máximo;
- Retransmissão dos dados, em caso de falha, no próximo período de serviço [5];

Este tipo de dados não tem uma estrutura de dados definida para a sua transmissão. Por outro lado, as ligações criadas para este tipo de comunicação são unidireccionais, o que faz que em caso de comunicação bidireccional é necessária a criação de duas ligações, uma para cada sentido.

Como no tráfego isócrono, também aqui o *endpoint* define qual a quantidade de dados máxima que pretende permitida para cada ligação criada. Este valor está limitado a 1024 bytes, sendo que tal como no caso das comunicações isócronas, também aqui,

exceptuando os *high-bandwidth endpoints*, podem enviar até 3 tramas por *microframe*, os *endpoints* estão limitados a 1 transacção por *microframe*.

Além do número de bytes a transmitir em cada transacção os *endpoints* podem definir também qual é o valor máximo que pretendem ter para o período de acesso ao barramento. A forma de definição deste período é idêntica à usada pelas ligações isócronas e é dada pela expressão, $(2^{bInterval-1}) * F$, onde *bInterval* pode assumir valores de 1 a 16 e F, assume o valor da duração de uma *microframe*, 125µs [5]. Tendo em conta isto, um *endpoint* que suporte tráfego de interrupções pode transmitir até 3072 bytes por *microframe*, ou seja, transmitir a uma taxa máxima de 192Mb/s.

3.4.4 Tráfego *Bulk*

Normalmente este tipo de tráfego é usado por aplicações que necessitem de transmitir grandes quantidades de informação, e em que esta possa transmitida apenas quando houver largura de banda disponível no barramento, dependendo assim a rapidez da comunicação da actividade do barramento em cada momento. Por outro lado, este tipo de comunicação garante a entrega correcta dos dados através de mecanismos de retransmissão de tramas e controlo de erros ao nível do hardware.

Aqui os dados não são transmitidos numa estrutura definida e as ligações são sempre unidireccionais, sendo necessária a criação de novas ligações caso se pretenda comunicação bidireccional.

Endpoints que utilizem este tipo de tráfego, não podem reservar largura de banda nem especificar o tempo máximo de acesso ao meio, dando-se a transmissão deste tipo de dados apenas quando há largura de banda disponível no barramento, sendo que caso exista também a necessidade da transmissão de tráfego de controlo este será primeiramente transmitido. Por outro lado, ao contrário de para as tramas de controlo, para este tipo de tráfego o *host* não reserva nenhuma percentagem de tempo para transmitir este tipo de tráfego, de maneira que nem o software cliente nem o *endpoint* nunca sabem quando vai ser transmitido este tráfego. Por outro lado, para o USB2.0, a quantidade de dados transferidos por trama é limitada a 512 bytes.

Capítulo 4 Análise de Requisitos

Como foi dito anteriormente, neste trabalho pretendia-se desenvolver uma plataforma que permitisse o controlo e supervisão de sistemas embebidos, baseada num PC e no protocolo de comunicação USB. Foi também especificado, que o sistema referido fosse capaz de monitorizar e controlar um conjunto, de pelo menos, 16 entradas e 16 saídas digitais, e 8 canais de entrada analógicos.

Neste capítulo, serão especificadas e justificadas as opções tomadas para a implementação da plataforma pretendida, com especial ênfase nas opções tomadas em relação ao protocolo a utilizar para interligação dos dispositivos de campo, à escolha do microcontrolador utilizado para controlar o sistema e a forma como este se conectaria ao barramento USB, e à escolha dos periféricos utilizados para o sistema poder cumprir com as funcionalidades requeridas.

4.1 Protocolo de interligação dos dispositivos de Campo

A escolha do protocolo a utilizar para a comunicação entre os dispositivos periféricos e o microcontrolador é de elevada relevância para este projecto pois condiciona de forma decisiva a escolha da arquitectura global do sistema, assim como, a escolha dos dispositivos a utilizar. Para esta escolha, havia uma condicionante, que se centrava na ponderação do uso do protocolo I²C para a implementação desta comunicação. Assim sendo, para uma tomada de decisão foram estudados o funcionamento do protocolo I²C, e de dois protocolos alternativos, o protocolo SPI e o protocolo CAN. Esta análise limitou-se a estas 3 alternativas, pois após uma análise sobre as soluções existentes, considerou-se que apenas estas reuniam, as características ao nível da performance e em termos de ligação física, que cumpriam os requisitos exigidos para a plataforma pretendida. Seguidamente será apresentada uma descrição geral do funcionamento de cada uma das alternativas e justificada a opção tomada.

4.1.1 Protocolo I²C

O protocolo I²C foi originalmente desenvolvido pela Philips em 1996. Este protocolo utiliza duas linhas para fazer a transmissão de dados, sendo estas a linha SCL, linha de transmissão do sinal de relógio, e SDA, linha de transmissão de dados. As duas linhas são bidireccionais estando as duas ligadas a resistências de *pull-up*. Isto permite que não haja uma tensão obrigatória para a utilização deste barramento, dependendo esta da tensão de alimentação a que estão ligadas estas resistências, sendo os níveis de tensão mais utilizados os 5V e os 3.3V.

Os dispositivos ligados ao barramento I²C podem desempenhar dois papéis distintos, ou mestre ou escravo, sendo que é possível a existência de vários mestres numa mesma rede. A diferença entre estes dois papéis, está no facto, de apenas os mestres poderem controlar a linha SCL, ou seja, o sinal de relógio.

Neste protocolo, os dispositivos são identificados por um endereço único para cada um deles. Por norma, estes são representados utilizando 7 bits, o que devido a 16 destes estarem reservados, limita o número de dispositivos que se podem conectar à rede a 112. Numa versão mais recente, estes passaram a ser representados por 10 bits, o que aumenta o número máximo de nós conectados para 1024. Por outro lado, o número de dispositivos no barramento, não é limitado apenas pelo número de bits de endereçamento, mas também pela capacidade máxima do barramento, fixada em 400pF.

A versão mais utilizada deste protocolo (versão standard) permite velocidades de transmissão típicas de 100kbit/s, sendo que as versões mais recentes permitem velocidades que podem atingir os 400kbit/s (*fast mode*), 1Mbit/s (*fast mode plus*) e os 3.4Mbit/s (versão *High Speed*).

4.1.1.1 Protocolo

Neste protocolo, as comunicações são sempre iniciadas pelo mestre. Estas iniciam-se através do envio de uma trama de controlo. Essa trama é constituída pelo bit de *start*, transição descendente da linha de dados, SDA, quando a linha de relógio está a 1, seguido pelo endereço do escravo com que pretende comunicar, e no final um bit

onde o mestre indica o que pretende fazer, se ler ou escrever informação. Se o referido escravo estiver na rede este vai responder com um sinal de *acknowledge*, sinal representado pela colocação por parte do escravo da linha a 0V, indicando assim a recepção deste byte.

Em situações de escrita, o mestre deve enviar sequencialmente os dados pretendidos, respondendo o escravo no final de cada byte com o sinal de *acknowledge*. Por outro lado, em situações de leitura, a situação inverte-se.

No final da transmissão de toda a informação pretendida, o mestre pode tomar duas opções: terminar a comunicação enviando um bit de stop, ou caso pretenda iniciar uma nova transmissão, enviar novamente um bit de *start*.

4.1.1.2 Controlo de Fluxo de Dados

No protocolo I²C, está implementado um método de controlo do fluxo de dados. Este método consiste em o escravo, quando não tem capacidade de processar todos os dados enviados pelo mestre, tomar conta da linha de relógio, colocando a 0V, impedindo assim o mestre de continuar dados. Quando o escravo recuperar desta situação liberta a linha de relógio e o mestre continua com o envio de dados.

4.1.1.3 Controlo de acesso ao Meio

O suporte de uma estrutura multi-mestre implica a implementação de um mecanismo que permita regular situações de conflito no acesso ao meio. Por isso, neste protocolo, durante o envio do endereço do escravo, todos os mestres lêem o barramento, no final do envio de cada bit e comparam esse valor com o valor enviado. Caso, estes não sejam idênticos, o mestre detecta que existe um outro mestre a aceder ao barramento, deixando de enviar os dados desejados, ficando à espera de um bit de stop para tentar enviar novamente os dados.

4.1.1.4 Endereços

Os endereços no I²C podem ser representados por 7 ou 10 bits, sendo únicos para cada tipo de dispositivo. Existem dois tipos de endereços: endereços parcialmente fixos, onde uma parte deste é programável, o que permite a utilização de dispositivos do mesmo tipo ao mesmo tempo no barramento; e endereços totalmente fixos, endereços impossíveis de alterar, o que impossibilita o uso de dispositivos do mesmo tipo ao mesmo tempo no barramento. Esta situação pode ser ultrapassada através do uso de *multiplexers* permitindo a separação do barramento.

O uso de *multiplexers* permite também aumentar a capacidade máxima de permitida num barramento. Ao separar o barramento em diferentes braços, cada um destes pode suportar a capacidade máxima permitida num barramento, 400pF, permitindo assim aumentar o número máximo de dispositivos ligados ao barramento. O uso de *multiplexers*, tem apenas o senão de não permitir, ao mestre, o acesso a todos os braços do barramento ao simultaneamente. A capacidade máxima do barramento pode ser também aumentada recorrendo ao uso de buffers e repetidores, com a vantagem de que estes não separam o barramento possibilitando assim o acesso ao todo o barramento ao mesmo tempo.

Outra das dificuldades que se poderá encontrar neste barramento, é o facto de existir a hipótese de se pretender conectar a este, dispositivos que trabalham a voltagens diferentes, por exemplo dispositivos que trabalham a 5V e dispositivos mais recentes que trabalham a 3.3V e não toleram tensões de 5V. Esta dificuldade pode ser ultrapassada com a utilização de *switches*, que permitem a divisão do barramento em diferentes “braços”, onde se utilize tensões diferentes.

O uso de *switches* permite também aumentar a fiabilidade do barramento. Como o barramento é montado numa arquitectura “*open-drain*”, se um dispositivo avaria e coloca uma das linhas ou no nível baixo ou alto, o barramento fica inutilizado pois nenhum dispositivo o vai conseguir controlar. Sendo que o barramento só volta a ficar disponível quando o dispositivo for desligado. Com o uso de *switches* é possível manter o barramento a funcionar apesar de algum dispositivo não estar a funcionar, pois como o *switch* separa o barramento em diferentes componentes é possível desactivar o braço

que se encontra inoperacional, permitindo assim a continuação do funcionamento do resto do barramento.

4.1.2 Protocolo SPI

O barramento SPI (*Serial Peripheral Interface*) é um barramento de comunicação série, que permite o funcionamento no modo *full-duplex*. Este é constituído por 4 linhas, SCK (*Serial Clock*), MOSI (*Master Output Slave Input*), MISO (*Master Input Slave Output*) e SS (*Slave Select*). Por outro lado, este barramento funciona numa arquitectura mestre-escravo, em que o mestre controla todas as comunicações efectuadas.

A linha SCK, como o próprio nome indica, é linha utilizada para transmitir o sinal de relógio que vai controlar a velocidade com que se vai dar a comunicação, sendo este enviado pelo mestre. As linhas MOSI e MISO, são as linhas utilizadas para a transmissão dos dados, a primeira para permitir o envio de dados do mestre para o escravo, enquanto a segunda, é utilizada para permitir o envio de dados no sentido inverso.

Por último, a linha de *Slave Select* é usada pelo mestre para seleccionar o dispositivo com que pretende comunicar. A selecção do respectivo escravo, faz com que a partir desse momento, apenas este leia o conteúdo da linha MOSI e se necessário envie dados através da linha MISO. Este modo de funcionamento impede que neste barramento, sejam transmitidas mensagens em *broadcast*.

Este barramento, como já referido, funciona numa arquitectura mestre-escravo, sendo apenas o mestre a dar início a uma transmissão de dados. O primeiro passo para iniciar uma comunicação, é configurar a frequência do sinal de relógio para a frequência a que deseja que a esta se dê. As frequências utilizadas aqui podem atingir os 70MHz. Depois de configurado o sinal de relógio, o mestre deverá seleccionar o dispositivo com que pretende comunicar, colocando a respectiva linha de SS a 0. A partir deste momento, é iniciada a transmissão dos dados, sendo possível, durante um ciclo de relógio, a transmissão de dados tanto pela linha MOSI como também pela linha MISO, havendo assim uma comunicação em modo full-duplex.

A comunicação neste barramento pode ser descrita como uma troca de conteúdos entre 2 *shift registers*, ligados em anel. Ao longo da transmissão dos dados, há uma troca progressiva entre os dois conteúdos. No final de cada transmissão, cada um dos dispositivos envolvidos lê o conteúdo do seu *buffer*, repetindo-se o processo se existirem mais dados a transmitir. Se por outro lado, não houverem mais dados a transmitir o mestre termina o envio do sinal de relógio e desselecciona o escravo.

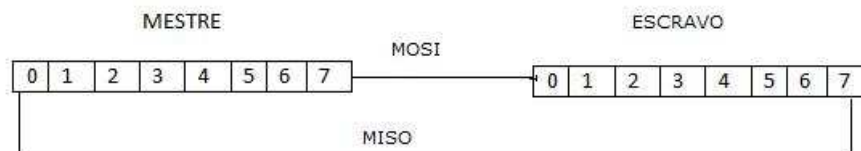


Figura 4- Troca de dados SPI

Para que se dê uma comunicação correcta dos dados, além de configurar a frequência do sinal de relógio, o mestre deverá configurar tanto a polaridade (CPOL) e a fase (CPHA) deste sinal relativamente aos sinais utilizados para a transmissão de dados. Esta configuração é essencial, pois irá determinar a forma como será gerado o sinal de relógio, definindo o seu valor base, CPOL, como também definirá o momento em que se deverão escrever e ler os dados, CPHA, estando estes associados a transições do sinal de relógio. Seguidamente, serão descritas as possíveis combinações do valor destas duas variáveis e o seu efeito nas transmissões:

Para CPOL = 0: valor base do sinal de relógio é zero;

- CPHA=0 – Os dados são lidos à transição positiva do sinal de relógio e devem ser escritos na transição negativa
- CPHA=1 – Os dados são lidos à transição negativa do sinal de relógio e devem ser escritos na transição positiva;

Para CPOL=1: valor base do sinal de relógio é zero;

- CPHA=0 – Os dados são lidos à transição negativa do sinal de relógio e devem ser escritos na transição positiva;
- CPHA=1 – Os dados são lidos à transição positiva do sinal de relógio e devem ser escritos na transição negativa;

Como foi dito anteriormente, o barramento SPI permite a conexão de vários escravos ao barramento. Normalmente, a topologia usada para fazer esta conexão é utilizar sinais de SS (*Slave Select*) independentes para cada um dos escravos,

controlados pelo mestre. Esta solução implica que o pino onde é ligada a linha MISO nos escravos seja de alta impedância pois esta linha é partilhada por todos os escravos.

Existe também uma outra topologia possível para a interligação de diferentes escravos no barramento. Esta topologia, chamada “*Daisy Chain*”, consiste na interligação do pino de MISO de um determinado escravo, com o pino MOSI do escravo adjacente. Quando usado este tipo de topologia, num primeiro conjunto de ciclos do sinal de relógio, os escravos recebem os dados enviados pelo mestre, colocando-os na linha MISO, no conjunto seguinte, passando assim a informação ao escravo seguinte. Desta forma toda a rede funciona como um único *shift register*.

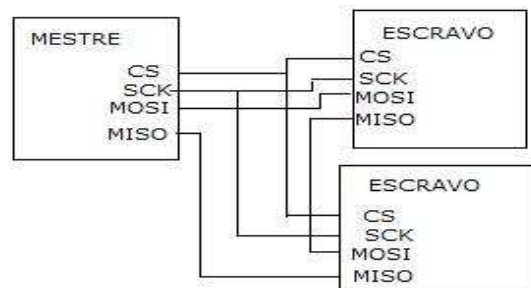


Figura 5- SPI topologia “*Daisy Chain*”

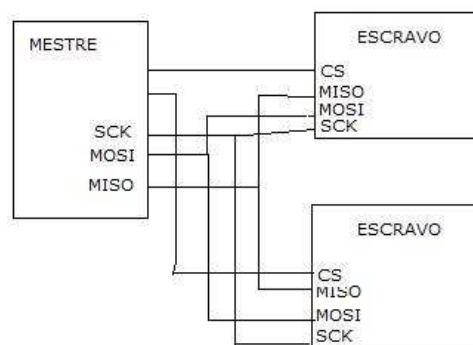


Figura 6- SPI topologia comum

4.1.3 Protocolo CAN

O CAN (*Control Area Network*) é um protocolo de comunicação série que foi criado para possibilitar a comunicação entre diferentes dispositivos sem necessidade de utilização de um computador para o efeito. Inicialmente, este foi implementado para utilização na indústria automóvel, de forma a interligar todos os constituintes dos sistemas de controlo destes. Com o passar do tempo, este foi passando a ser utilizado em outras áreas de controlo.

Neste barramento, são utilizadas apenas duas linhas para a transmissão de dados, sendo esta diferencial, com velocidades de transmissão até 1Mbit/s para distâncias inferiores a 40 metros e de 125Kbit/s para distâncias até 500 metros.

Aqui, a comunicação de dados é feita recorrendo ao envio de mensagens com formato predefinido, variando apenas o tamanho de cada uma destas. Neste protocolo, não são utilizados endereços para identificar os dispositivos, sendo o encaminhamento das mensagens efectuado recorrendo ao campo que identifica o seu tipo. Assim, cada dispositivo é configurado inicialmente, com os tipos de mensagem a que deverá prestar atenção. Este método, além de permitir a transmissão de dados em *multicast*, permite também que não seja necessário que cada um dos nós tenha informação sobre o sistema, facilitando assim a adição de novos dispositivos à rede.

Num barramento CAN, todos os nós estão em pé de igualdade, sendo então possível, que quando pretender, cada um deles possa aceder ao meio para a transmissão de dados. Para permitir este funcionamento, foi necessário criar um mecanismo que gerisse o acesso ao meio, resolvendo possíveis conflitos. O mecanismo criado baseia-se na implementação de um sistema de prioridades dos diferentes tipos de mensagens. A sua implementação consiste em, durante a transmissão do campo que indica o tipo de mensagem, cada nó, depois de enviar um bit, leia o barramento e verifique se o seu valor é idêntico ao valor do bit que enviou. Caso isso não aconteça, o nó deverá parar de transmitir dados, pois existem mais nós a aceder ao barramento, simultaneamente, transmitindo mensagens com maior nível prioridade.

4.1.4 Escolha do Protocolo a utilizar

A rede de comunicação escolhida para interligar o microcontrolador aos periféricos foi a SPI. Essa escolha baseou-se essencialmente no facto de esta rede permite velocidades mais elevadas, permitir comunicação full-duplex e permitir uma maior flexibilidade na implementação do protocolo de comunicação, pois como não existe um protocolo definido a liberdade era total para desenvolver um protocolo que otimiza-se a comunicação entre os dispositivos conectados ao barramento. Por outro lado, a solução SPI apresenta como grande desvantagem, em relação às outras soluções ponderadas, o facto de utilizar mais linhas para a transmissão de dados. Principalmente o facto de ser usada uma linha de *Slave Select* para cada um dos escravos ligados à rede, o que pode ser limitativo para o número de dispositivos a ligar à rede, aspecto que assumia especial importância, visto que deveria ser contemplada, no projecto inicial a hipótese de poderem ser conectados mais dispositivos à rede.

Outra vantagem do SPI em relação ao I²C, é ter uma interface de hardware extremamente simples, sem necessidade de resistências de *pull-up* e com menos requisitos de potência. Tem também a vantagem de não ser necessário haver mecanismos de controlo de acesso ao meio, não havendo assim a hipótese de existirem possíveis erros associados a este tipos de mecanismos.

Por outro lado, a opção SPI tem como desvantagens não ter controlo de fluxo por hardware, não existir pré definido *acknowledge* por parte do escravo, o que faz com que o mestre não saiba se as tramas enviadas foram recebidas. Esta desvantagem pode ser ultrapassada implementando um mecanismo de *acknowledge* manual no protocolo de comunicação desenvolvido. Outras desvantagens são o facto de a existência de diferentes mestres simultaneamente ser muito complicada e não haver hipótese de haver comunicação *multicast*. Uma outra desvantagem comparativamente ao CAN é a distância que a rede pode atingir, mas que para a solução pretendida não era um aspecto de grande relevância.

4.2 Escolha do microcontrolador

Escolhido o protocolo a utilizar, o próximo passo concentra-se na escolha dos microcontrolador a utilizar. O microcontrolador terá um papel fulcral em todo o sistema desenvolvido pois é o responsável pela recolha e controlo dos dados de modo a poder ser implementado o sistema de SCADA. Para a escolha do microcontrolador foi imposto um condicionalismo inicial que se centrou no tipo de encapsulamento deste, sendo que se optou por não utilizar um dispositivo com encapsulamento do tipo SMD, pois a plataforma desenvolvida é um protótipo e pretendia-se utilizar dispositivos que pudessem ser testados mesmo antes de se desenvolver uma placa de circuito impresso.

Na escolha do microcontrolador, havia um ponto importante que condicionaria a escolha deste, a forma como seria feita a conexão ao barramento USB. Neste ponto existiam duas hipóteses, a primeira era utilizar um microcontrolador que tivesse a interface USB integrada, a segunda opção consistia na utilização de um dispositivo que fizesse a interface entre o protocolo USB e o protocolo RS-232, através do qual se conectaria o microcontrolador. Para a escolha se tornar um pouco mais simples, principalmente devido ao elevado número de fabricantes, foi decidido utilizar um dispositivo da marca Atmel, pois é um fabricante conceituado e com provas dadas em aplicações industriais.

Depois de especificada a marca de dispositivo a utilizar foram analisadas as características dos dispositivos que continham interface USB e que cumprissem as características de encapsulamento pretendido. Depois de analisadas as características dos dispositivos com interface USB e comparando-as com as dos dispositivos que não a contêm, conclui-se que a melhor opção seria utilizar um conversor USB RS-232 para fazer a conexão ao barramento USB. Esta escolha baseou-se no facto de os microcontroladores com interface USB não conterem canais analógicos suficientes, o que iria obrigar à utilização de um periférico que disponibilizasse os 8 canais pretendidos, sendo que esse periférico seria controlado através do barramento SPI, o que aumentaria muito o tráfego do barramento, assim como a complexidade da gestão deste. Outro aspecto importante para a decisão tomada, foi o facto de, caso tivesse sido escolhido um dispositivo com interface USB teria que ser feita a interface física do microcontrolador ao barramento, sendo necessário calcular a ferrite a utilizar nesta

interface o que aumentaria a dificuldade do projecto da placa de circuito impresso produzida. Assim depois de analisadas todas as opções, foi decidido utilizar o dispositivo ATMEGA644 e o kit UM-232 da FTDI, como conversor USB RS-232.

4.3 Escolha dos periféricos

Como foi dito, pretendia-se que o sistema possibilita-se o controlo e a monitorização de 16 entradas e 16 saídas digitais e de 8 entradas analógicas. Este último ponto foi resolvido utilizando os canais analógicos fornecidos pelo microcontrolador, ficando apenas por resolver o problema das entradas e saídas digitais. Para resolução desta questão, decidiu-se usar então extensões de portas digitais.

A escolha dos dispositivos baseou-se em 3 parâmetros essenciais: a possibilidade de comunicação por SPI, o número de portas fornecidas por cada dispositivo e a possibilidade dos dispositivos, usados para suportar as entradas, poderem sinalizar mudanças de estado destas, de uma forma rápida e que simplificasse o processo de leitura por parte do microcontrolador.

Um dos pontos analisados na escolha dos dispositivos a utilizar, foi o número de portas fornecidas por este. Este parâmetro é bastante importante, pois pretendia-se cumprir os requisitos necessários usando o menor número de dispositivos possível, de forma a tornar o controlo de todas as comunicações mais simples e limitar o menos possível a capacidade de expansão deste sistema, sendo que devido ao facto de no SPI a linha de *chip select* é independente para cada dispositivo, quantos mais dispositivos usados para cumprir os requisitos, menos portas do microcontrolador sobriam para possíveis expansão, limitando de sobremaneira esta.

Por último, foi analisada também a capacidade dos dispositivos em sinalizar a ocorrência de mudanças de estado dos sinais de entrada de forma rápida e eficaz. Esta capacidade é de extrema relevância para permitir um processo de monitorização eficaz, pois se não existisse esta capacidade, a leitura teria de ser feita de forma cíclica pelo microcontrolador, o que poderia levar a uma utilização pouco eficaz do barramento SPI. Assim como, levaria ao aumento, de forma bastante importante, das possibilidades de perda de possíveis eventos de curta duração.

Depois de analisadas várias alternativas existentes no mercado, dando especial relevância aos critérios especificados anteriormente, foi escolhido o dispositivo MCP23S17. Este dispositivo cumpre todos os requisitos pretendidos, fornecendo um conjunto de 16 portas que podem ser configuradas tanto como entradas como saídas, possibilitam a geração de um sinal de interrupção, configurável, quando existe um evento nas portas.

4.4 Resumo da solução Proposta

Depois de analisados todos os requisitos do sistema e ponderando todas as hipóteses possíveis, a solução final proposta é constituída pela utilização do microcontrolador ATMEGA644, o kit UM-232 da FTDI como conversor USB-RS232 e as extensões de portas digitais da *Microchip*, MCP23S17. Com a selecção proposta, os limites máximos do sistema ao nível das velocidades máximas de transmissão possíveis, situam-se nos 10MHz na comunicação via SPI, máximo permitido tanto pelos periféricos como pelo microcontrolador, se usado relógio a 20MHZ. Ao nível da comunicação via USB-RS232, como foi decidido utilizar o dispositivo como se fosse uma porta série normal, o limite máximo da velocidade das comunicações situar-se-á no limite permitido pela aplicação utilizada para fazer a interface com os operadores no computador visto que o conversor permite uma velocidade máxima de 1MBaud e o microcontrolador, a 20MHz, permite velocidades que podem atingir os 2.5Mbit/s.

Com a solução proposta, pretende-se implementar um sistema, em que a sua arquitectura funcional seria idêntica à arquitectura de um sistema de SCADA, sendo esta apresentada na figura 7.

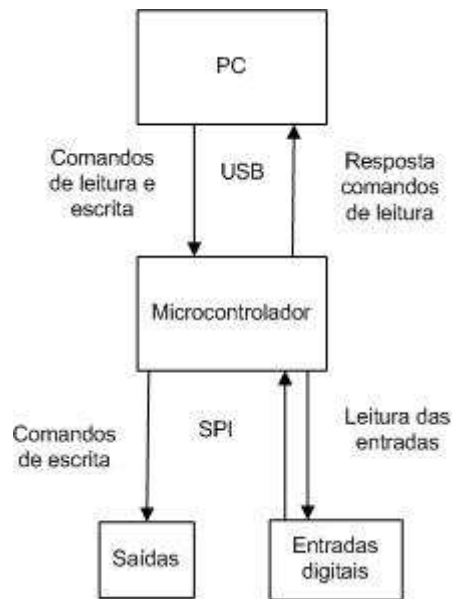


Figura 7- Arquitectura Funcional do Sistema

Analisando a figura 7, podemos ver que este sistema tem 2 elementos essenciais: a aplicação que se encontra no computador e o microcontrolador. A primeira tem como função permitir a interacção do sistema com o utilizador, permitindo-lhe visualizar o estado das entradas e o envio de comandos de escrita nas saídas do sistema. Por outro lado, o microcontrolador vai ser o centro do sistema. Este é responsável pela leitura e escrita efectiva do valor das entradas e saídas do sistema, sendo também responsável pela recepção e resposta aos comandos de leitura e escrita enviados pelo utilizador.

Capítulo 5 Implementação

Depois de no capítulo anterior ter sido enunciado e justificado as escolhas para a implementação do sistema pretendido, neste será feita uma descrição da sua implementação e do seu funcionamento. Começar-se-á por se fazer uma descrição da implementação a nível de hardware e depois fazer-se-á a descrição do funcionamento ao nível do software.

Como qualquer sistema de SCADA, o trabalho produzido, é constituído por duas partes essenciais: uma placa de circuito impresso, onde estão o microcontrolador e os periféricos utilizados para fazer a aquisição dos dados e controlo do sistema, e por uma aplicação no computador que permite ao utilizador interagir com o hardware utilizado, permitindo-lhe ler o valor das entradas e também controlar o valor das saídas. De notar que a plataforma produzida, não está vocacionada, neste momento, para nenhum sistema específico, sendo que a aplicação do computador serve apenas para teste da comunicação via USB e teste do funcionamento em geral.

5.1 Circuito Produzido

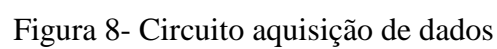
Nesta secção, será apresentada uma descrição geral do circuito projectado para suportar a plataforma, com a descrição da função dos seus principais constituintes, seguida de uma descrição mais pormenorizada de alguns dos pontos mais importantes, como por exemplo, a interface com os sinais a adquirir.

O circuito projectado pode ser dividido em duas partes: uma primeira parte, que é utilizada para fornecer os requisitos mínimos ao trabalho, ou seja, permite a aquisição de dados e a comunicação com o computador; a segunda parte, é utilizada para cumprir um outro objectivo, que é possibilitar outro método de visualização do que está a acontecer, através de um LCD. Isto é feito através da utilização de mais dois microcontroladores, um dedicado unicamente à interface com o LCD, e um outro que tem como função controlar os dados que irão aparecer neste, sendo responsável também pela sua obtenção através da comunicação com o microcontrolador principal. Além da possibilidade de interacção com o sistema via LCD, foi incluída a possibilidade dos

utilizadores poderem interagir com o sistema através de um pequeno teclado, que poderá ser utilizado para configurar alguns aspectos do sistema. À que notar que a implementação, ao nível do software, da interface com o teclado e o LCD não foram devidamente implementadas, devido principalmente a motivos temporais.

Relativamente à parte de aquisição e controlo de entradas e saídas, há a salientar que além da utilização dos componentes essenciais, microcontrolador, extensões de portas e conversor USB-RS232, foram também utilizados componentes cuja função é proteger os mecanismos de aquisição de dados de má utilização por parte dos operadores. Outro facto importante, é que pretendia-se criar um sistema expansível por isso foi incluído também um conector extra que permite a conexão de 8 dispositivos via barramento SPI.

Um ponto importante no projecto deste circuito, era como seria feita a sua alimentação. Existiam de início duas hipóteses: ou se utilizaria a alimentação fornecida pelo barramento USB, ou se utilizaria uma fonte de alimentação externa. Para a decisão entre estas duas hipóteses foram tidos em consideração essencialmente dois critérios: primeiramente, a potência necessária para alimentar todo o circuito, e segundo os níveis de tensão que seriam necessários utilizar. Tendo em conta estes dois critérios, optou-se pela utilização de uma fonte externa. Esta escolha justifica-se, por o barramento USB no máximo fornecer uma corrente de 500mA, que poderia ser limitativa para o sistema proposto, assim como pelo facto de serem necessários níveis de tensões superiores a 5V, o que implicaria utilizar um sistema que permitisse a subida dos níveis de tensão para pelo menos 12V. A fonte utilizada fornece ao sistema tensões de 5, 15 e -15V contínuas, sendo depois utilizados dois rectificadores para conseguir, a partir do 15V, os 12 utilizados para alimentar os relés, e os 3.3V a partir dos 5V.



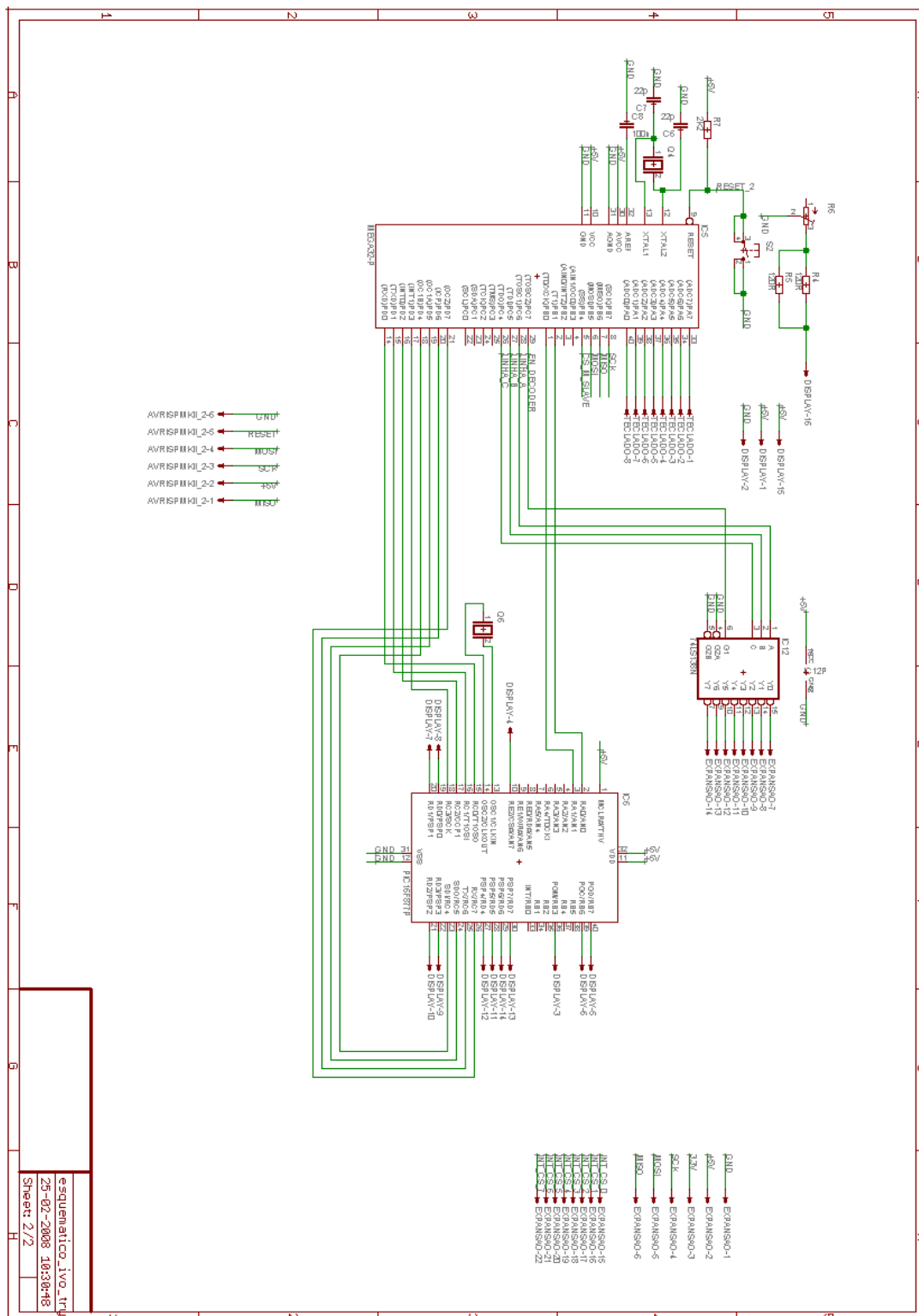


Figura 9- Circuito de controlo do LCD

5.1.1 Entradas e Saídas digitais

Como foi dito anteriormente, no projecto da placa de circuito impresso foram incluídos dispositivos que permitissem proteger as entradas do sistema, tanto analógicas como digitais, em casos de má utilização deste. Ao nível das entradas digitais, esse perigo centrava-se na aplicação de níveis de excessivos nas extensões de portas que funcionam como entradas, que poderiam levar à sua danificação. Para que isso não acontecesse, foram colocados antes destes dispositivos, *buffers* digitais, dispositivo SN74LVCZ245ADW, que garantem que o valor das suas saídas não ultrapassará o valor da alimentação, neste caso 5V, valor próprio para trabalhar com os periféricos utilizados. Além de concederem alguma protecção, a utilização destes dispositivos permite aumentar a estabilidade dos sinais que chegam às entradas do sistema, ponto importante, para reduzir as possibilidades de existirem alterações do valor lógico induzidas por ruído.

No caso das saídas digitais, foram também utilizados os mesmos dispositivos. Neste caso, não tanto por motivos de protecção, mas pela possibilidade que este dispositivos possuem de funcionarem a níveis de tensão diferentes. Facto este, que permite que os sinais de 5V, vindos do dispositivo que controla as saídas digitais, sejam traduzidos para outros níveis de tensão. Aqui pretendia-se que o sistema fosse capaz de interagir com sistemas que funcionassem tanto a 5V como a 3.3V, o valor inicial é 3.3V. Para conseguir este objectivo, era necessário montar um sistema que permitisse a troca da alimentação dos *buffers* de forma automática. Este sistema foi constituído, recorrendo ao uso dos dispositivos G6H2-100 e ULN2803. O primeiro é um relé que é utilizado para permitir a comutação entre os dois níveis de tensão, sendo que a saída de um dos seus contactos está ligado directamente à alimentação do buffer, permitindo assim através da sua comutação a troca do valor da tensão de alimentação. O dispositivo ULN2803 é um conjunto de oito transístores par de *darlington* utilizados para controlar a comutação do relé. Ao ser aplicado um sinal de 5V, numa entrada, o respectivo transístor vai conduzir permitindo que sejam aplicados os 12V nos terminais do relé, necessários para a sua comutação. De notar que os *buffers* têm um sinal de *chip enable*, que permite inibir as suas saídas, o que permite que durante o processo de comutação estas sejam desligadas. O sinal é controlado pelo microcontrolador. Neste sistema, foi também colocado, em anti-paralelo com o relé, um diodo 1N4148, de forma que este

conduza quando o relé está contra polarizado protegendo-o assim. Foi também colocado, um led, em paralelo com o relé, para permitir de forma simples saber qual a tensão seleccionada, 5V com o led aceso 3.3V com este desligado.

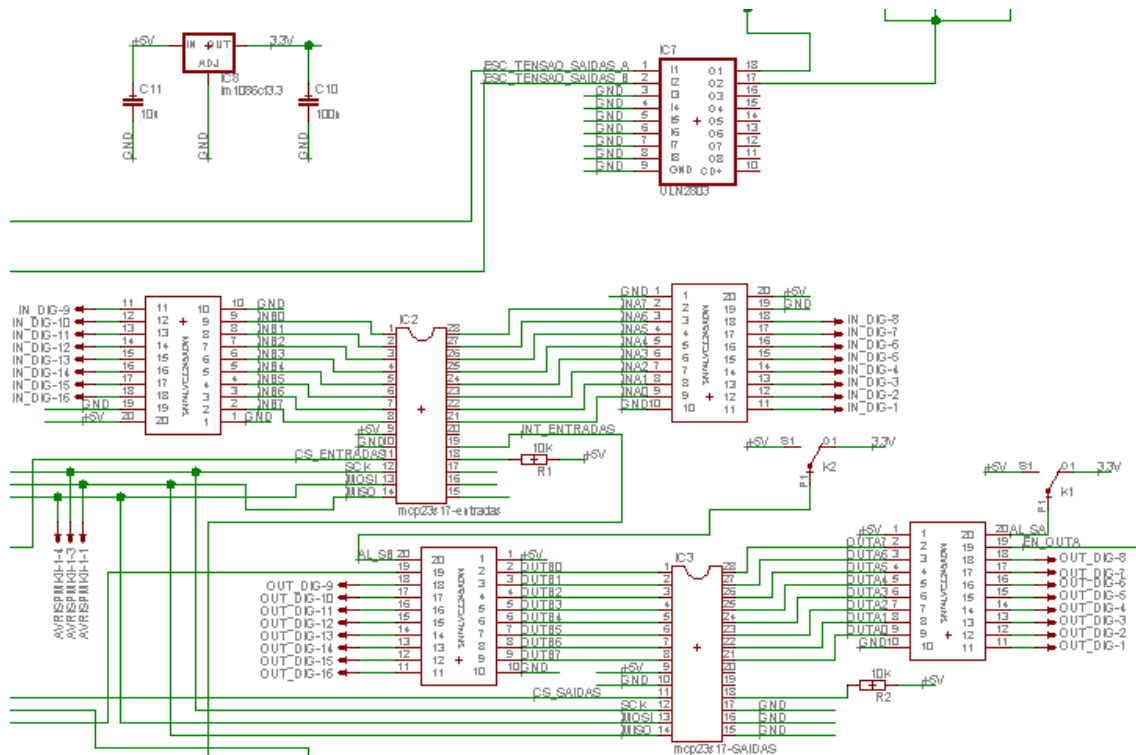
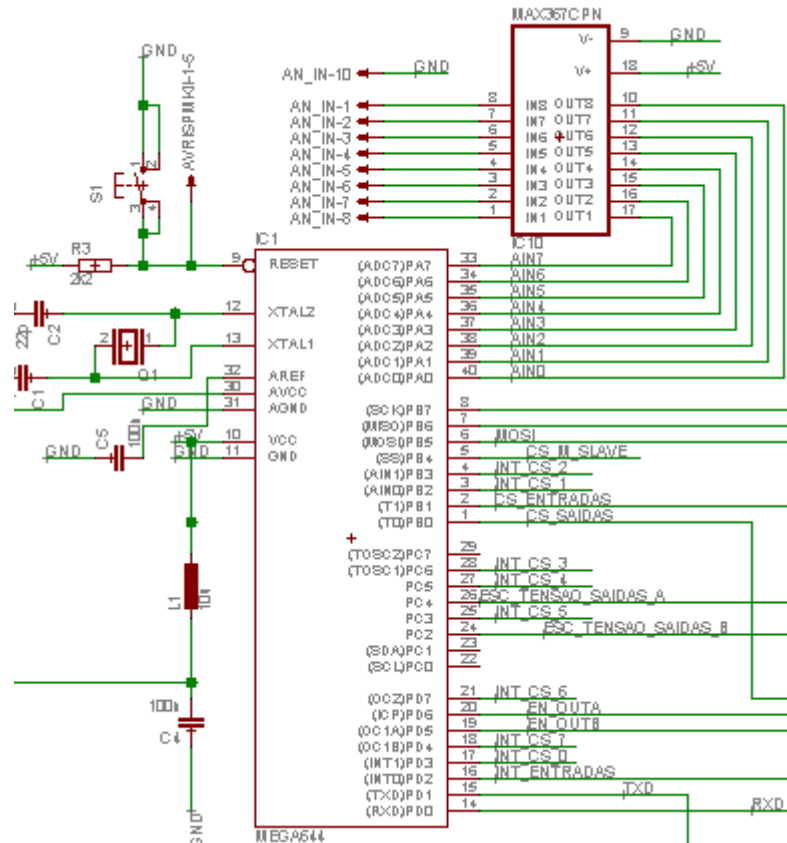


Figura 10- Entradas e Saídas Digitais

5.1.2 Entradas Analógicas

Relativamente, às entradas analógicas, pretendia-se garantir que qualquer que fosse a tensão colocada nos sinais a monitorizar, estas nunca saíssem da gama de 0 a 5V, gama de funcionamento dos canais analógicos existentes no microcontrolador. Com este objectivo, foi utilizado o dispositivo MAX367. Este dispositivo é feito para permitir a protecção de circuitos contra sobre tensões. Este dispositivos funciona como um conjunto de 8 resistências, cujo valor é variável e que garante que a tensão das suas saídas não sai da sua gama de alimentação. Em termos práticos, cada conjunto de entrada saída é ligado utilizando um conjunto de 3 transístores *mosfet*, dois do tipo N e um do tipo P, que vão controlando a corrente que passa entre a entrada e a saída de forma que a tensão de saída nunca saí da sua gama de alimentação. A gama de alimentação utilizada na alimentação deste componente foi de 0 a 5V. A gama escolhida

deveria ter sido um pouco mais larga, pois devido às regras de condução dos transístores, a gama onde a tensão de entrada é idêntica à tensão de saída é mais pequena que os 0 a 5V.



incluídos sinais de alimentação de 5 e 3.3V, que poderão ser utilizados para alimentar os novos dispositivos.

Para conseguir os 8 sinais de *chip select* necessários para suportar a comunicação SPI, foi utilizado um *decoder*, o dispositivo SN74HC138N. Este dispositivo foi utilizado por possibilitar o controlo de 8 sinais recorrendo ao uso de apenas 4, 3 para seleccionar o sinal a activar e um a funcionar como *enable*. Devido à escassez de pinos disponíveis no microcontrolador principal, este dispositivo é controlado pelo microcontrolador secundário, que recebe do primeiro a informação sobre qual sinal deve activar através do barramento SPI.

5.2 Microcontrolador

O microcontrolador é o elemento central de todo o sistema, fazendo a ponte de ligação entre todos os seus constituintes. Este tem como função fundamental fazer a aquisição dos sinais de entrada, tanto digitais como analógicos, e controlar o valor das saídas do sistema. Além do mais, tem que disponibilizar a informação recolhida ao operador, possibilitando à aplicação do computador o acesso a essa informação através da comunicação via USB. Tem que permitir também, que o operador possa manipular as saídas do sistema, através dos mesmos meios.

Apesar de ser o centro do sistema, o seu funcionamento é puramente reactivo, pois como neste sistema pretendia-se apenas a leitura das entradas e a escrita nas saídas e não foi implementada nenhuma função de controlo autónoma, o que faz que a execução destas funções fique dependente da ocorrência de determinados eventos. Estes eventos podem ser, o envio de mensagens por parte do computador a pedir a disponibilização do valor das entradas ou a escrita nas saídas, ou por exemplo a ocorrência de uma interrupção que indique que houve mudanças no valor das entradas. Assim o funcionamento do microcontrolador pode ser melhor compreendido através do seguinte fluxograma.

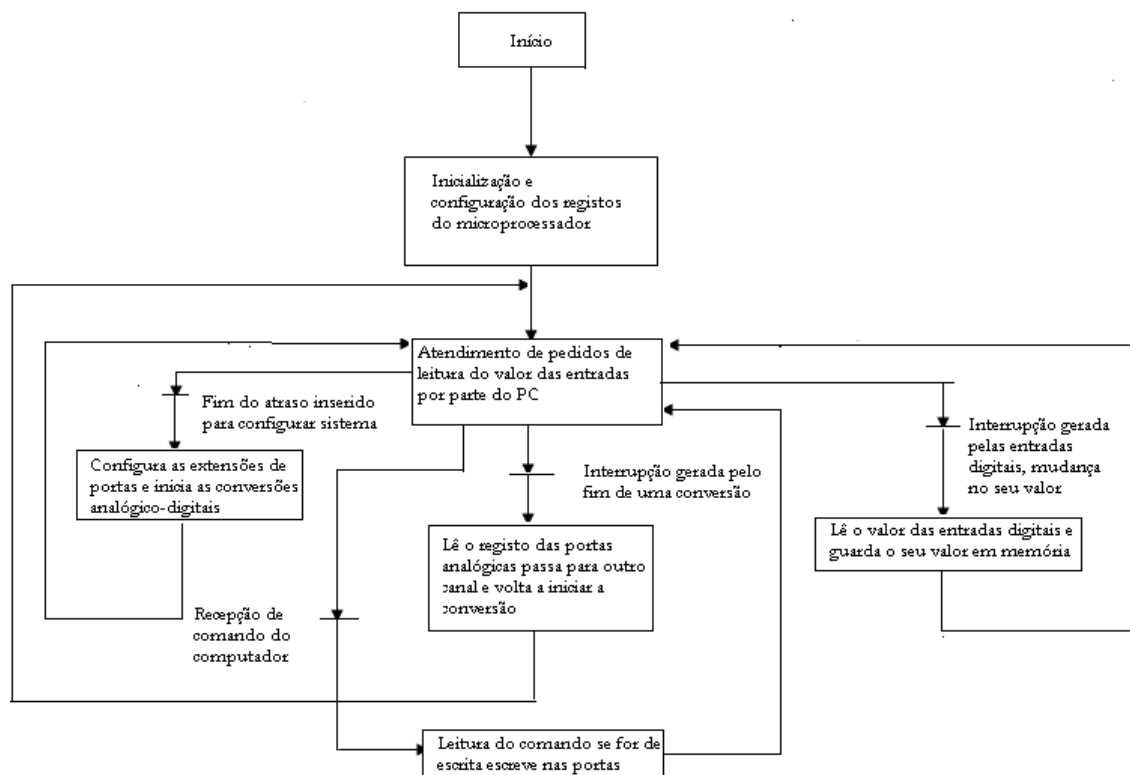


Figura 12 - Fluxograma do funcionamento do microcontrolador

Ao analisar o fluxograma inserido na imagem anterior, pode-se constatar o comportamento puramente reactivo do microcontrolador, baseando-se este apenas na resposta à ocorrência de eventos. Estes eventos podem ser essencialmente de 3 tipos: recepção de comandos vindos do computador, geração de interrupções por parte das entradas digitais indicando mudanças no seu valor e interrupções geradas pelos conversores analógico digitais. No fluxograma pode-se visualizar a ocorrência de um outro evento, relacionado com a introdução de um pequeno atraso para ser feita a configuração das extensões de portas e o início das conversões dos canais analógicos. Este atraso foi introduzido, devido a alguma dificuldade em obter resposta dos dispositivos MCP23S17, caso as rotinas de interrupção fossem executadas no processo de configuração normal do microcontrolador. O valor do atraso introduzido foi de 1ms.

Como foi dito anteriormente, um dos eventos que geram resposta por parte do microcontrolador é a recepção de comandos vindos do computador. O envio dos comandos por parte do computador é feita via o conversor USB-RS232 utilizado. Para comunicação destes comandos, foi desenvolvido um protocolo de comunicação cujo funcionamento é descrito pormenorizadamente na secção seguinte. Estes comandos

podem ser divididos em 3 tipos distintos: comandos de configuração, comandos de escrita nas saídas e comandos de leitura de dados. Os comandos de configuração são utilizados pelo computador para obter informação sobre quantos bits são usados na conversão dos sinais analógicos ou para mudar a velocidade de transmissão utilizada na comunicação entre este e o microcontrolador. Por sua vez, os comandos de escrita nas saídas são utilizados pelo computador para indicar ao microcontrolador qual os valores que devem ser escritos nas saídas digitais, sendo que a escrita nas saídas digitais é feita utilizando a comunicação SPI com a extensão de portas respectiva.

O outro tipo de mensagens enviadas pelo computador indica que este pretende aceder aos valores das entradas do sistema, tanto digitais como analógicas. Estes valores encontram-se guardados na memória do microcontrolador e reflectem o valor das entradas no momento da última leitura. Para o tratamento deste tipo de comandos, foi criada uma lista, onde são guardados todos os pedidos pendentes que são respondidos por ordem de chegada. Esta lista foi criada para não sobrecarregar o código implementado na rotina de recepção de dados vindos da porta série, além do facto da resposta a estes pedidos não ter requisitos temporais tão apertados como a resposta ao comando de escrita.

Uma das outras funções do microcontrolador é a aquisição do valor das entradas analógicas conectadas ao sistema. A aquisição destas entradas é feita recorrendo aos 8 canais analógicos existentes no microcontrolador, sendo estes lidos de forma consecutiva. O sinal de relógio utilizado, para controlar a velocidade com que são feitas as conversões foi de 1MHz, sendo que cada conversão demora 13 ciclos de relógio com mais um ciclo de intervalo entre estas. Este tempo, entre conversões, na realidade será um pouco maior pois como foram usados os oito canais, é necessário acrescentar a este tempo, o tempo gasto na configuração do canal a utilizar. Sem contar este tempo, a conversão de todos os canais seria feita, em aproximadamente 112 μ s.

Os canais analógicos existentes, permitem o uso de 10 bits nas conversões, mas como se pretendia que o sistema apenas que permitisse detectar variações significativas nos sinais medidos, foi tomada a opção de se apenas utilizar 8 dos 10 bits disponibilizados. Os 8 bits utilizados foram os mais significativos, o que tornou os resultados das conversões disponibilizados menos sujeitos a pequenas variações.

Por último, o microcontrolador tem também de ler o valor das entradas digitais. A leitura destes valores é feita através da comunicação SPI, utilizando o protocolo desenvolvido pelo fabricante, que será descrito posteriormente. Este processo é desencadeado pela ocorrência de uma interrupção externa no microcontrolador. Essa interrupção é gerada pela mudança de sinal do pino do periférico, configurado para que os seus pinos funcionem como entrada, que é usado por este para indicar ao utilizador que ocorreu uma mudança no valor de um dos seus pinos, permitindo assim que não seja necessária a constante leitura do valor das entradas para detectar mudanças.

5.3 Protocolo de Comunicação Série

Como já foi anteriormente referido, para fazer a comunicação entre o computador e o microcontrolador, optou-se pela utilização de um conversor USB-RS232, de modo a que permitisse a escolha de um microcontrolador que não tivesse embutido o interface USB, sendo que tanto para o computador como para o microcontrolador, a comunicação acontece como se fosse uma porta série normal.

Para conseguir uma comunicação fiável e segura entre o computador e o microcontrolador, foi necessário desenvolver um protocolo de comunicações que permitisse implementar todo o tipo de comunicações desejáveis de forma simples e segura. Para isso, foi desenvolvida uma trama que permitisse o encapsulamento dos dados a transmitir. Na figura 12, é especificado o formato dessa trama.

Cabeçalho	Tipo	Dados	FCS
-----------	------	-------	-----

Figura 13-Trama de dados série

Apenas o campo “Dados” tem tamanho variável, tendo todos os outros campos, tamanho fixo, sendo este de um byte para o campo “cabeçalho” e “FCS”, e de dois para o campo “Tipo”. O primeiro campo tem como objectivo indicar o início de uma trama, assumindo o valor de “H”. Por outro lado, o último campo é utilizado para verificação do conteúdo recebido. Este campo é preenchido pelo emissor com o valor do ou exclusivo de todos os outros bytes enviados na trama. Na recepção, por outro lado, o

receptor depois de receber toda trama faz a mesma operação até ao penúltimo byte lido, comparando depois o valor desta operação com o último byte lido, o que permite caso sejam diferentes detectar erros de transmissão e assim desprezar os dados recebidos.

No campo “Dados” são enviados os dados relativos a cada tipo de mensagem, podendo estes ser valores das portas digitais, valores de portas analógicas, ou valores de tensão a aplicar nas saídas, entre outros.

Por outro lado, no campo “Tipo” é especificado, como o nome indica, qual o objectivo da mensagem enviada. Na tabela seguinte, serão especificados todos os tipos de mensagens utilizadas e a sua função.

Tabela 1- Descrição da função das tramas porta série

Tipo	Descrição
IN	Esta mensagem é enviada pela aplicação no computador para poder saber quantos bits estão a ser utilizados pelo microcontrolador na conversão dos sinais analógicos, com o objectivo de em futuras aplicações, caso seja aumentado o número de bits utilizados na conversão, não ser necessário modificar o software de recepção da trama do valor lido nos canais analógicos
RA	Esta mensagem é utilizada pelo computador para ler o valor das portas analógicas. No envio do pedido a aplicação, especifica as portas que pretende ler. Esta especificação é feita usando apenas um byte, sendo que cada bit desse byte representa cada uma das 8 portas. Caso se pretenda ler uma determinada porta, por exemplo a dois, o segundo bit deverá estar activo, funcionando de maneira idêntica para as restantes portas. Na resposta, há que notar que o microcontrolador envia o valor de todas as portas cujo valor foi pedido, começando na com endereço mais pequeno e seguindo sempre a forma ascendente.
RD	Com o envio desta mensagem a aplicação do computador pretende ler o valor das entradas digitais. Ao enviar esta mensagem o computador especifica qual o dispositivo que pretende ler, utilizando um byte que pode assumir o valor entre 0 e 8. Na resposta o microcontrolador responde utilizando dois bytes, um onde é indicando o valor das 8 entradas com número mais baixo e outro onde indica o valor das 8 entradas mais altas. Na resposta é indicado também o endereço do dispositivo correspondente.
WD	Este comando é utilizado pelo computador para manipular as saídas digitais. Aqui, além de especificar qual o valor que cada uma das portas deve assumir, indica qual dos dispositivos em que pretende escrever. No caso de ser o dispositivo base, dá também, informação sobre a tensão de alimentação de cada um dos buffers associados às saídas, sendo que o valor de '0' indica a tensão de 3.3V e o valor de '1' do valor de 5V. De notar que por porta, designa-se cada conjunto de 8 saídas, representando assim a maneira como estão divididos os pinos na extensão de portas.
BR	Mensagens deste tipo são utilizadas pela aplicação do computador para indicar que o utilizador pretende mudar a velocidade de transmissão da comunicação via porta série. Ao enviar este comando passa como parâmetros o valor com que o microcontrolador deve configurar os registos de configuração da <i>baudrate</i> . Quando é enviada esta mensagem a aplicação deixa de enviar mensagens via porta série até receber uma mensagem de retorno do microcontrolador indicando-lhe que recebeu a mensagem.

De notar que neste protocolo, todas as comunicações são iniciadas pelo envio de comandos por parte da aplicação no computador aos quais poderá corresponder uma resposta do microcontrolador.

O facto de, para um determinado tipo de comando, a única diferença entre o comando enviado pelo computador e a resposta enviada pelo microcontrolador, se situar no campo dos dados e no seu tamanho, permite que a rotina de recepção de mensagens via porta série seja implementada de forma idêntica dos dois lados da comunicação. Esta rotina é implementada utilizando uma máquina de estados, que tem por base o formato das tramas especificado anterior. Na figura 13 é especificado, o funcionamento da máquina de estados implementada.

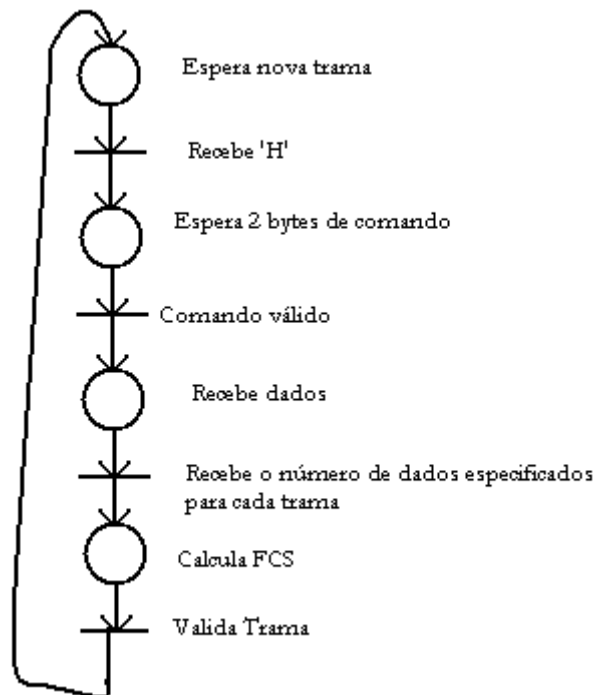


Figura 14-Máquina de estados série

5.4 Protocolo Comunicação SPI

O barramento SPI é fundamental no funcionamento de todo o sistema. É através deste que é feita a interacção do microcontrolador com as extensões de portas e também com o microcontrolador auxiliar que controla o funcionamento do LCD. Para o funcionamento correcto das comunicações via SPI, o aspecto talvez mais importante, é o controlo dos sinais de *chip select*, que permitem ao microcontrolador seleccionar o dispositivo com que pretende comunicar. Por isso mesmo, a primeira acção efectuada sempre que se pretende usar o barramento SPI, é activar todos os sinais de *chip select*, desseleccionando assim todos os dispositivos, de modo a depois seleccionar o periférico pretendido.

Em relação à interacção com os dispositivos MCP23S17, apenas foi necessário implementar no microcontrolador rotinas que permitissem cumprir o protocolo especificado na *datasheet* do dispositivo para leitura e escrita dos registos de memória deste. Por outro lado, para a comunicação entre os dois microcontroladores, tal como na comunicação com o computador, foi necessário fazer a especificação de um protocolo que permitisse a comunicação entre os dois. Nas secções seguintes, serão descritos o funcionamento de cada um dos respectivos protocolos.

É importante referir que, devido à arquitectura mestre-escravo inerente ao funcionamento do barramento SPI, esse papel é assumido pelo microcontrolador principal.

5.4.1 Comunicação com as extensões de portas

Para a comunicação com o MCP23S17, extensão de portas digitais, apenas foi necessário seguir o protocolo especificado pelo fabricante. Aspecto importante a reter na explicação do funcionamento deste protocolo é que estes dispositivos apenas assumem o papel de escravos dentro do barramento SPI, respondendo apenas a comandos enviados pelo mestre, neste caso o microcontrolador. Importante notar

também que com a comunicação SPI, o microcontrolador está apenas a aceder ou manipular o conteúdo de um conjunto de registos que permitem configurar o funcionamento dos dispositivos e controlar também esse funcionamento.

O protocolo especificado pelo fabricante para interagir com estes dispositivos é bastante simples. Consiste no envio de um primeiro byte onde o mestre indica se pretende escrever ou ler nos dispositivos, seguido do endereço do primeiro registo cujo ao conteúdo pretende aceder. Depois, caso pretenda escrever no registo, o mestre deve enviar o valor que pretende que ele assuma, caso pretenda ler deverá apenas continuar com o envio de sinais de relógio e esperar pela resposta do dispositivo.

O byte de comando que inicia as comunicações é constituído pelo endereço do dispositivo e por um bit que indica que tipo de acção se pretende efectuar. O endereço do dispositivo é constituído pelos 4 bits mais significados, que assumem o valor de 0100, e por mais 3 bits que poderão ser configurados pelo utilizador via hardware recorrendo aos pinos 15, 16 e 17 de cada dispositivo. Por outro lado, o utilizador pode também configurar se o valor desses pinos é tido ou não em conta pelo dispositivo, ficando o endereço com o valor de 0x40 se tal acontecer. No último bit, o mestre indica o que pretende fazer, se ler ou escrever, sendo que caso o seu valor seja 0 indica que pretende escrever nos registos, caso seja 1 pretende ler.

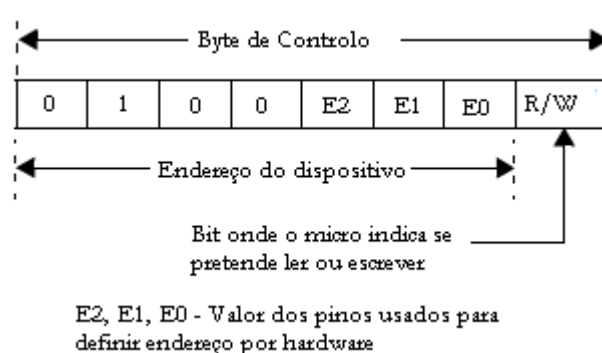


Figura 15-Byte de comando comunicação Periféricos

Como foi dito anteriormente, na comunicação com as extensões de portas digitais, o microcontrolador está apenas a aceder a um conjunto de registos. Como as 16 portas existentes estão organizadas em dois conjuntos de oito, essa divisão nota-se também nos registos existentes, visto que cada um dos conjuntos tem os próprios registos, permitindo uma configuração independente de cada uma das portas. Por outro

lado, estes dispositivos permitem configurar como estão organizados cada conjunto de registos. Existem então duas possibilidades de configurar essa organização: a primeira, onde cada conjunto é organizado sequencialmente, de forma que os registos da porta A têm endereços consecutivos e só no final destes é organizada os registos da porta B de igual forma; a segunda possível maneira de organizar os registos, consiste em os registos com a mesma função de cada uma das portas assumirem endereços consecutivos.

Register Name	Address (hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	POR/RST value
IODIRA	00	IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0	1111 1111
IODIRB	01	IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0	1111 1111
IPOLA	02	IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0	0000 0000
IPOLB	03	IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0	0000 0000
GPINTENA	04	GPINT7	GPINT6	GPINT5	GPINT4	GPINT3	GPINT2	GPINT1	GPINT0	0000 0000
GPINTENB	05	GPINT7	GPINT6	GPINT5	GPINT4	GPINT3	GPINT2	GPINT1	GPINT0	0000 0000
DEFVALA	06	DEF7	DEF6	DEF5	DEF4	DEF3	DEF2	DEF1	DEF0	0000 0000
DEFVALB	07	DEF7	DEF6	DEF5	DEF4	DEF3	DEF2	DEF1	DEF0	0000 0000
INTCONA	08	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	0000 0000
INTCONB	09	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	0000 0000
IOCON	0A	BANK	MIRROR	SEQOP	DISSLW	HAEN	ODR	INTPOL	—	0000 0000
IOCON	0B	BANK	MIRROR	SEQOP	DISSLW	HAEN	ODR	INTPOL	—	0000 0000
GPPUA	0C	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	0000 0000
GPPUB	0D	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	0000 0000
INTFA	0E	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	0000 0000
INTFB	0F	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	0000 0000
INTCAPA	10	ICP7	ICP6	ICP5	ICP4	ICP3	ICP2	ICP1	ICP0	0000 0000
INTCAPB	11	ICP7	ICP6	ICP5	ICP4	ICP3	ICP2	ICP1	ICP0	0000 0000
GPIOA	12	GP7	GP6	GP5	GP4	GP3	GP2	GP1	GP0	0000 0000
GPIOB	13	GP7	GP6	GP5	GP4	GP3	GP2	GP1	GP0	0000 0000
OLATA	14	OL7	OL6	OL5	OL4	OL3	OL2	OL1	OL0	0000 0000
OLATB	15	OL7	OL6	OL5	OL4	OL3	OL2	OL1	OL0	0000 0000

Register Name	Address (hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	POR/RST value
IODIRA	00	IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0	1111 1111
IPOLA	01	IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0	0000 0000
GPINTENA	02	GPINT7	GPINT6	GPINT5	GPINT4	GPINT3	GPINT2	GPINT1	GPINT0	0000 0000
DEFVALA	03	DEF7	DEF6	DEF5	DEF4	DEF3	DEF2	DEF1	DEF0	0000 0000
INTCONA	04	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	0000 0000
IOCON	05	BANK	MIRROR	SEQOP	DISSLW	HAEN	ODR	INTPOL	—	0000 0000
GPPUA	06	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	0000 0000
INTFA	07	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	0000 0000
INTCAPA	08	ICP7	ICP6	ICP5	ICP4	ICP3	ICP2	ICP1	ICP0	0000 0000
GPIOA	09	GP7	GP6	GP5	GP4	GP3	GP2	GP1	GP0	0000 0000
OLATA	0A	OL7	OL6	OL5	OL4	OL3	OL2	OL1	OL0	0000 0000
IODIRB	10	IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0	1111 1111
IPOLB	11	IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0	0000 0000
GPINTENB	12	GPINT7	GPINT6	GPINT5	GPINT4	GPINT3	GPINT2	GPINT1	GPINT0	0000 0000
DEFVALB	13	DEF7	DEF6	DEF5	DEF4	DEF3	DEF2	DEF1	DEF0	0000 0000
INTCONB	14	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	0000 0000
IOCON	15	BANK	MIRROR	SEQOP	DISSLW	HAEN	ODR	INTPOL	—	0000 0000
GPPUB	16	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	0000 0000
INTFB	17	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	0000 0000
INTCAPB	18	ICP7	ICP6	ICP5	ICP4	ICP3	ICP2	ICP1	ICP0	0000 0000
GPIOB	19	GP7	GP6	GP5	GP4	GP3	GP2	GP1	GP0	0000 0000
OLATB	1A	OL7	OL6	OL5	OL4	OL3	OL2	OL1	OL0	0000 0000

Figura 16-Tabelas que reflectem a organização dos registos [6]

O dispositivo permite também configurar o modo como são acedidos os registos. Assim existem também, aqui, duas hipóteses de configuração: uma primeira, onde o

apontador para os diferentes registos não é incrementado internamente pelo dispositivo, fazendo com que ao ler, por exemplo, ao enviar ciclos de relógio extra continuamente se vá aceder sempre ao mesmo registos, ou se vá alternando entre a leitura dos dois registos idênticos de cada uma das portas, que acontece quando os registos se encontram seguidos; por outro lado, na segunda configuração possível, o endereço a ler é automaticamente incrementado pelo dispositivo, o que permite aceder aos diferentes registos apenas enviando continuamente ciclos de relógio.

A configuração dos dois aspectos enunciados anteriormente teve bastante importância na implementação do protocolo de comunicação. Assim, os registos foram configurados para que os endereços dos correspondentes em cada uma das portas tivessem endereços consecutivos e para que o endereço dos registos fosse automaticamente incrementado pelo dispositivo. Estas duas configurações foram feitas de forma a otimizar a comunicação minimizando a quantidade de dados que seria necessário transmitir, mas também para facilitar a implementação do protocolo ao nível do software. Assim com as configurações escolhidas as acções de leitura e escrita ficaram limitadas à transmissão de 4 bytes, byte de comando, byte do endereço do registo e os dois bytes com o conteúdo dos registos. Por outro lado, o benefício desta escolha sente-se principalmente na acção de configuração dos dispositivos, pois o facto destes, por defeito, ao ligar-se a alimentação, estarem configurados desta maneira, tornou o processo mais simples e mais rápido.

Um dos aspectos referido para justificar a escolha deste dispositivo, foi a possibilidade de ser gerada uma interrupção sempre que detectado uma mudança de estado de um dos pinos de entrada. Foi necessário também fazer escolhas na forma como seriam geradas as interrupções. O MCP23S17 tem dois sinais de interrupção, um por cada conjunto de 8 pinos, podendo estes sinais funcionar de duas maneiras distintas. Na primeira, cada um deles funciona de forma independente do outro, reflectindo no seu valor a ocorrência de eventos que geram interrupções no seu conjunto de pinos. Na segunda configuração possível, estes podem estar conectados internamente, sinalizando assim, ambos, interrupções correspondentes a qualquer uma das portas. Para não obrigar à ligação de mais sinais aos pinos do microcontrolador, o que limitaria muito a expansão do sistema, optou-se por configurar os sinais de interrupção para reflectirem eventos ocorridos em ambas as portas.

Outro aspecto importante no processo de geração de interrupções é a configuração do motivo que leva à ocorrência de uma interrupção. Este dispositivo permite gerar interrupções por dois motivos diferentes, ou por o valor do pino diferir do valor pré-definido, ou seja, se for pré-definido que o valor normal do pino é 5V só é gerada interrupção se o valor for colocado a 0V, ou por haver apenas uma mudança no valor do pino, ou seja, independentemente do valor *default*, se houver uma alteração no valor do pino é gerada uma interrupção.

Quando gerado um sinal de interrupção, este é mantido activo até à leitura de um de 2 registos. Um desses registos é o registo que reflecte o valor dos pinos em cada momento, GPIO_n, com n A ou B. O outro registo é o registo INTCAP. Este registo reflecte o valor das portas no momento em que foi gerada a interrupção. Nos testes realizados ao sistema, optou-se por quando ocorresse uma interrupção fosse lido o registo GPIO, permitindo visualizar o valor das portas em cada momento e não momento da interrupção, o que facilita a avaliação da rapidez do processo de leitura. Por outro lado, o dispositivo foi configurado para gerar interrupções sempre que houvesse uma mudança no valor lido nos pinos de entrada.

5.4.2 Interacção entre os dois microcontroladores

Além de ser utilizado para permitir a interacção entre o microcontrolador principal e as extensões de portas digitais, o barramento SPI é utilizado para fazer a comunicação entre o microcontrolador principal e o microcontrolador secundário. Esta comunicação é utilizada, pelo primeiro, para controlar os sinais de *chip select* disponibilizados no conector de expansão e para permitir a interacção com o LCD.

Para implementar as funcionalidades referidas, foi utilizado um protocolo de comunicações extremamente simples. As mensagens utilizadas são constituídas por apenas dois campos, um que indica o tipo de comando e um segundo que contém os dados a ser transmitidos, não sendo obrigatório a existência de um campo de dados.

Tipo	Dados
------	-------

Figura 17-Trama SPI comunicação entre dois micros

O campo de “Tipo” é enviado utilizando apenas um byte, enquanto o campo “Dados”, não tem comprimento fixo. Neste campo, são enviados, ou o número do sinal de *chip select* a activar, ou os comandos a enviar para o LCD.

No protocolo desenvolvido foram criados 3 tipos de comandos: um para ser indicado qual o sinal de *chip select* a activar, outro para desactivar todos os sinais de *chip select*, e, por último, um comando que permite enviar comandos para o LCD. Na tabela seguinte, serão especificados as características de todos os comandos utilizados.

Tabela 2-Tipo de comandos usado na comunicação entre os dois microcontroladores

Tipo	Descrição
1	Este comando é utilizado para indicar qual o sinal de <i>chip select</i> que deverá ser desactivado. Ao receber este comando o microcontrolador secundário vai manipular os sinais que permitem a sua interacção com o <i>decoder</i> , de modo a apenas a saída desejada ser desactivada. Permitindo assim a comunicação com o periférico respectivo.
2	O comando do tipo 2 é utilizado para permitir a activação de todos os sinais de <i>chip select</i> do conector de expansão. O envio deste comando vai levar à desactivação do sinal de <i>enable</i> do <i>decoder</i> , fazendo com que todos os sinais de saída do <i>decoder</i> fiquem activos. Neste comando, é enviado apenas o byte que indica o tipo de comando, não sendo enviados quaisquer dados.
3	Este comando é utilizado para passar comandos que depois serão enviados para o LCD. Esses comandos são enviados utilizando o campo de dados. Devido a serem utilizados comandos de tamanho diverso na interacção com o LCD, o primeiro byte do campo de dados é utilizado para indicar qual o tamanho dos dados a ser enviado.

A maneira como são recolhidos os dados para serem visualizados no LCD, poderia ser feita de maneira diferente. Em vez do microcontrolador principal, controlar directamente o conteúdo do LCD, com o envio de comandos o tipo 3, este controlo poderia ser feito exclusivamente pelo microcontrolador secundário, tornando assim, completamente independente o controlo do LCD do controlo do resto do sistema.

Este controlo poderia ser feito, implementando, no microcontrolador secundário, rotinas que permitissem que este pudesse aceder aos dados recolhidos pelo

microcontrolador principal por iniciativa própria. O acesso a esses dados seria feito, utilizando igualmente, o barramento SPI. Isto implicaria, que devido à arquitectura do barramento, onde não existe a possibilidade da existência em simultâneo de vários mestres, houvesse uma troca temporária de papéis entre os dois microcontroladores. Esta troca é possível de implementar, utilizando a funcionalidade do microcontrolador, que através da manipulação do valor do pino SS, *chip select* do microcontrolador, permite a troca automática de mestre para escravo e possibilita a leitura imediata dos dados enviados.

A utilização deste método de controlo do conteúdo do LCD seria talvez mais eficaz do que o implementado, mas devido escassez de tempo para a sua implementação, e por não ser essencial para os objectivos pretendidos para este trabalho, a sua implementação não foi feita. Por outro lado, a implementação deste método, poderia provocar uma degradação da performance do sistema de aquisição de dados, devido à perda do controlo do barramento SPI, por parte do microcontrolador principal, que posteriormente teria que ser avaliada.

5.5 Interacção com o LCD

Como foi anteriormente referido, para a comunicação com o LCD são utilizados dois microcontroladores, um responsável unicamente pela implementação da interface com o LCD e um outro responsável pelo controlo dos dados que vão ser exibidos. Este microcontrolador é também responsável pela obtenção dos dados a mostrar, através da comunicação com o microcontrolador principal, comunicação descrita na secção anterior.

Depois de obter os dados que pretende exibir, o microcontrolador tem de os transmitir para o microcontrolador responsável pela interface com o LCD de forma a este saber o que tem e como tem que exibir. Para implementar desenvolveu-se um protocolo, que é implementado recorrendo a uma interface paralela, sendo usado 8 pinos para a transmissão dos dados, mais 2 para controlo das comunicações.

Para possibilitar uma comunicação simples e de fácil implementação, as mensagens do protocolo definido foram estruturadas de uma forma extremamente

simples. Assim estas são constituídas por dois campos fundamentais: o campo “Comando” e o comando “Dados”. Na figura 18, é apresentada a estrutura das mensagens.

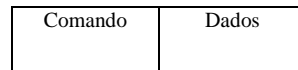


Figura 18- Formato do Comandos para escrever no LCD

O campo “Comando” é representado apenas por um byte e indica qual a função que se pretende executar. Por outro lado, o campo “Dados” tem tamanho variável e é usado para passar os parâmetros da função a executar. Neste campo, são passadas informações, como as coordenadas onde se executará a função, os dados a escrever, a cor com que serão escritos, entre outros.

Como já foi referido, esta comunicação utiliza 10 pinos, 8 para passar os dados constituintes dos comandos, mais 2 para controlar o fluxo da informação. Desses dois pinos, um é controlado pelo microcontrolador que pretende escrever algo no LCD e o outro é controlado pelo microcontrolador que vai receber os dados para depois inseri-los neste. Assim, a transferência dos dados é processada da seguinte forma: de início os 2 sinais de controlo estão ambos desactivados, quando o primeiro microcontrolador pretende transmitir dados, coloca o valor do byte que quer transmitir nas 8 linhas de dados, depois de fazer isso, activa o sinal de controlo que controla, indicando que pretende transmitir dados, ao se aperceber deste sinal, o outro microcontrolador vai ler o conteúdo das linhas de dados, e quando ler esse conteúdo, este activa o seu sinal de controlo indicando que leu os dados, ao receber este sinal o emissor desactiva o seu sinal de controlo, ficando depois à espera da desactivação do outro sinal de controlo por parte do receptor, indicando que os dados foram processados e pode receber outro byte. No envio de um comando, este procedimento repete-se para a transmissão de todos os bytes constituintes da trama. O processo descrito é representado na figura 19.

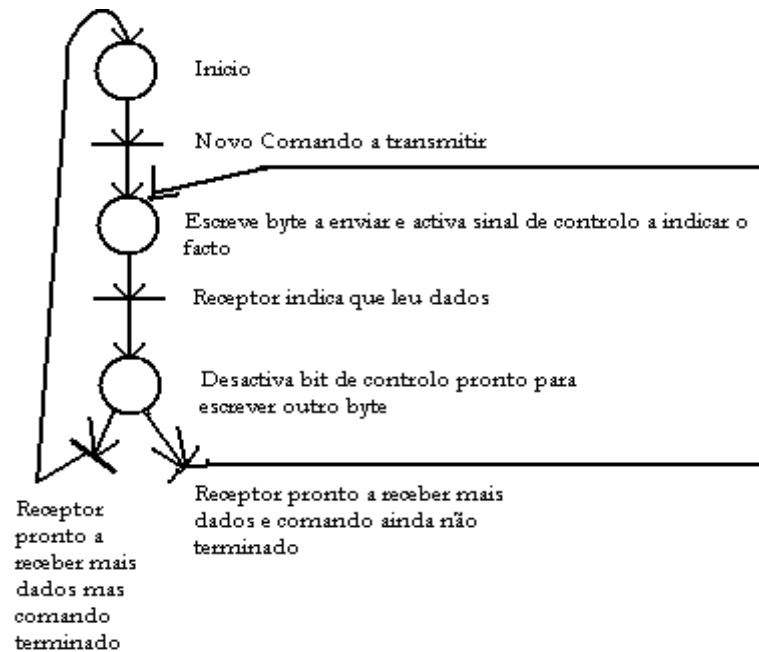


Figura 19-Máquina de estados da comunicação entre os dois micros para controlar LCD visto do emissor

5.5 Aplicação do Computador

Para testar o funcionamento da plataforma desenvolvida integrada num sistema de SCADA, era obrigatório o desenvolvimento de uma aplicação no computador que permitisse implementar as funcionalidades básicas deste tipo de sistemas. Permitindo o acesso aos dados adquiridos e o controlo das saídas do sistema por parte do operador. É importante referir, que com esta aplicação se pretendia apenas testar as capacidades de aquisição de dados e controlo do sistema desenvolvido, não tendo sido desenvolvida nenhuma função de interpretação dos dados adquiridos, que permitisse a geração de alarmes, ou a criação de um histórico, que teriam de ser desenvolvidos caso fosse aplicado a um sistema de controlo concreto.

5.5.1 Plataforma de Desenvolvimento

Para servir de suporte à aplicação desenvolvida surgiram várias hipóteses. Entre essas hipóteses estavam, a criação de uma aplicação usando PHP e HTML, usar o C++ *Builder*, ou desenvolver uma aplicação usando o compilador de C, GCC.

O desenvolvimento da aplicação recorrendo à utilização do PHP tinha como principais vantagens, o facto de permitir o acesso remoto bastando para isso o recurso a um ponto de acesso à Internet, e de ser uma linguagem independente do sistema operativo, bastando assim apenas que o computador onde o sistema estivesse ligado possuísse uma ligação à Internet e uma porta USB para poder servir de interface para este sistema. Contudo, esta solução tem um grande senão que impossibilitou a sua utilização, o facto de o componente que permite a conexão via porta série, necessitar que fosse feita a configuração da porta série sempre que fosse carregada uma página, o que tornava o sistema extremamente lento, impossibilitando-a assim de cumprir as funções desejadas.

Outra das hipóteses ponderadas foi a criação de uma aplicação utilizando a linguagem de programação C e o compilador GCC. Como se pretendia, para já, desenvolver uma aplicação para o sistema operativo Windows, seria utilizada para a interacção com a porta série a API do Windows, mas visto que este compilador pode ser utilizado em diferentes sistemas operativos, facilmente poderiam ser desenvolvidas aplicações que permitissem a utilização em diferentes plataformas, como por exemplo, Linux. Esta solução provavelmente seria a melhor, mas devido ao facto, de não permitir a criação de uma interface gráfica de forma simples e rápida, que permitisse de uma forma simples a interactividade do utilizador, impossibilitou a sua utilização neste trabalho.

Por último, temos a solução C++ *Builder*. Esta foi a solução escolhida, visto que de forma simples e rápida, permite suplantar os problemas apontados às soluções acima referidas, a criação de uma interface gráfica e a implementação da comunicação série de forma funcional para os objectivos impostos. Por outro lado, a utilização desta solução é a única que não permite a migração entre sistemas operativos.

5.5.2 Funcionamento

Como foi dito, com o desenvolvimento desta aplicação apenas se pretendia testar o funcionamento do sistema, tendo esta como funcionalidades, apenas a visualização do estado das entradas do sistema e a possibilidade de controlo das suas saídas. Assim foi criado apenas um formulário, onde se pode visualizar o estado das entradas analógicas e o valor da tensão que está a ser medido em cada um dos canais analógicos, e seleccionar o valor que se pretende colocar nas saídas digitais bem com a sua tensão de saída.

Para testar o sistema foram criados dois modos de funcionamento: um modo manual, onde o utilizador tem controlo total sobre o sistema, através da manipulação de botões, e um modo de funcionamento automático, onde ciclicamente a aplicação escreve nas saídas do sistema e lê as entradas deste permitindo a simulação de diferentes situações de funcionamento.

No modo de funcionamento manual, como foi dito, a aplicação apenas responde a comandos do operador, que através da manipulação de *check boxes* e de botões pode controlar o estado das saídas, ler as entradas digitais e enviar comandos de leitura dos canais analógicos, podendo seleccionar quais deles pretende ler.

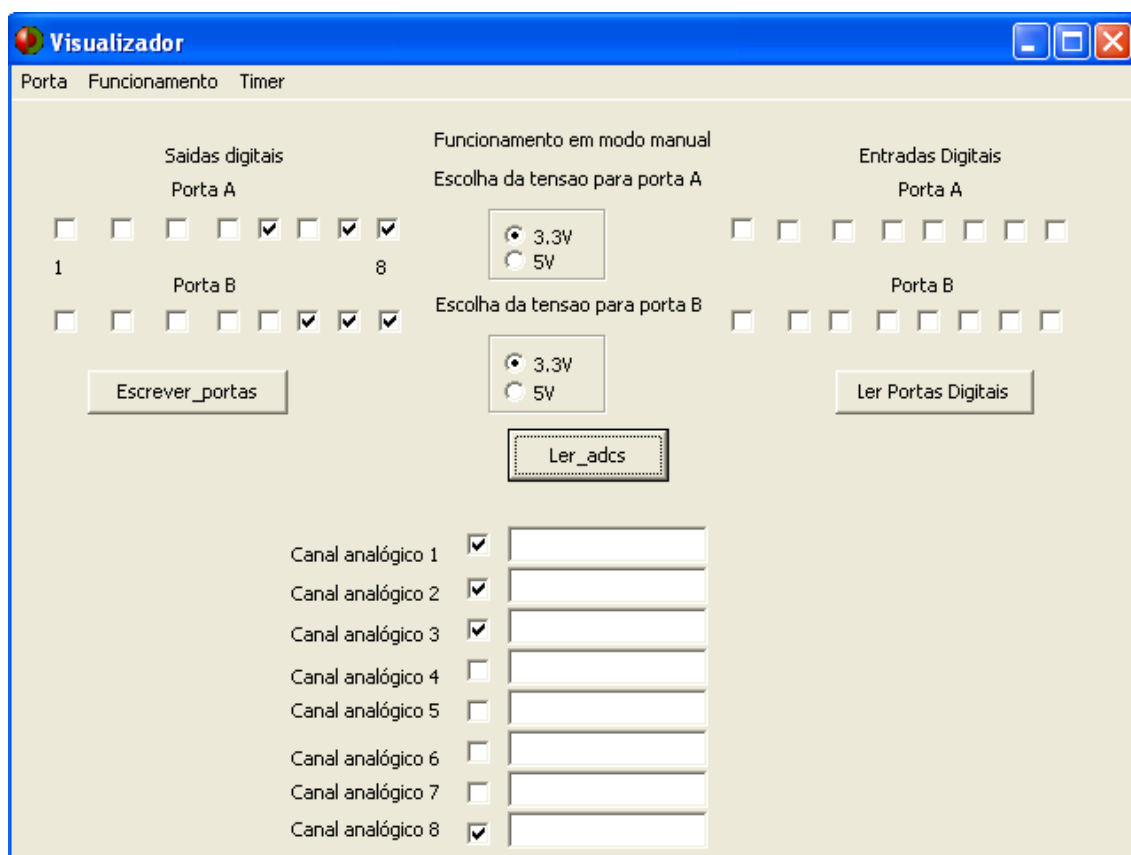


Figura 20- Funcionamento em modo manual

Em funcionamento automático, o operador pode apenas simular situações predefinidas do funcionamento do sistema. Neste modo, ciclicamente, é lido o valor das entradas, sendo também possível manipular o valor das saídas digitais. Este modo é especialmente útil para simular um funcionamento contínuo do sistema.

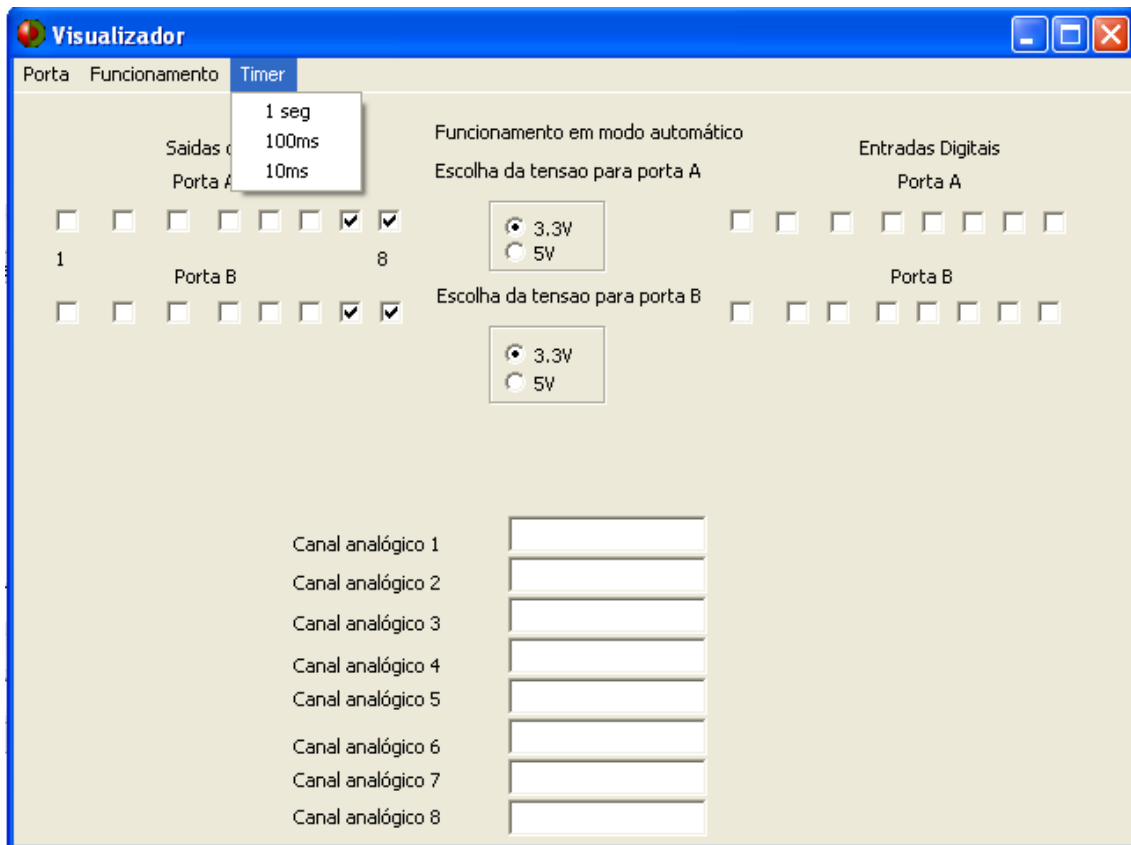


Figura 21-Funcionamento em modo automático

Para permitir a selecção e configuração dos diferentes modos de funcionamento foi criado um menu, que permite ao utilizador, entre outros, seleccionar o modo de funcionamento e as suas características. É possível através deste menu, controlar também alguns aspectos do funcionamento da porta série. Sendo possível controlar, o seu estado, conectado ou desconectado, e controlar a velocidade com que são as comunicações, com a escolha da velocidade de transmissão utilizada.

Quando selecciona o modo de funcionamento automático, o utilizador pode configurar as suas duas principais características, o intervalo de tempo em que se dá o envio dos comandos para o microcontrolador, podendo escolher entre tempos de ciclo de 1 segundo, 100ms ou 10ms, e o modo como se vão ser manipuladas as saídas podendo escolher, entre um modo onde o valor escrito para cada conjunto de 8 saídas é incrementado de 0 até ao valor de 255, e um outro modo, onde será mudado o estado das portas seleccionadas pelo utilizador, entre 0 e 1.

5.6 Resultados

Nesta secção serão apresentados alguns resultados, que permitem caracterizar o funcionamento do sistema desenvolvido, sendo apresentados dados que permitem caracterizar a velocidade de aquisição do estado das entradas digitais, assim como, a capacidade de processamento, por parte do sistema, de comandos de controlo e de leitura enviados pelo operador. Assim, para a realização destes testes, as saídas e entradas digitais do sistema foram ligadas entre si, permitindo assim testar em simultâneo, tanto as capacidades de aquisição de dados, como as capacidades de controlo, com o envio de comandos de escrita por parte do computador, seguidos de comandos de leitura que permitiam testar se os dados eram bem processados. Durante a realização, destes testes foi utilizado um sinal de relógio de 8MHz, permitindo uma velocidade de comunicação via SPI de 4Mbits/s. Por outro lado, as comunicações via porta série estavam configuradas para funcionar a 9600bit/s.

5.6.1 Velocidade de Leitura das Portas Digitais

Um dos testes que foi realizado serviu para medir o tempo que o microcontrolador demora a ler o valor das entradas digitais quando se dá a mudança de uma delas. Para isso, depois de executado o processo configuração das entradas e saídas digitais, foram activadas todas as saídas digitais, de modo a provocar uma interrupção que levaria à leitura das entradas por parte do microcontrolador. No final do atendimento da rotina da leitura, foi activado um dos pinos do microcontrolador de modo a sinalizar esse facto. O teste feito consistiu na medição do tempo decorrido entre a activação de uma das saídas e a activação desse pino por parte do microcontrolador. A medição obtida é apresentada na figura 22.

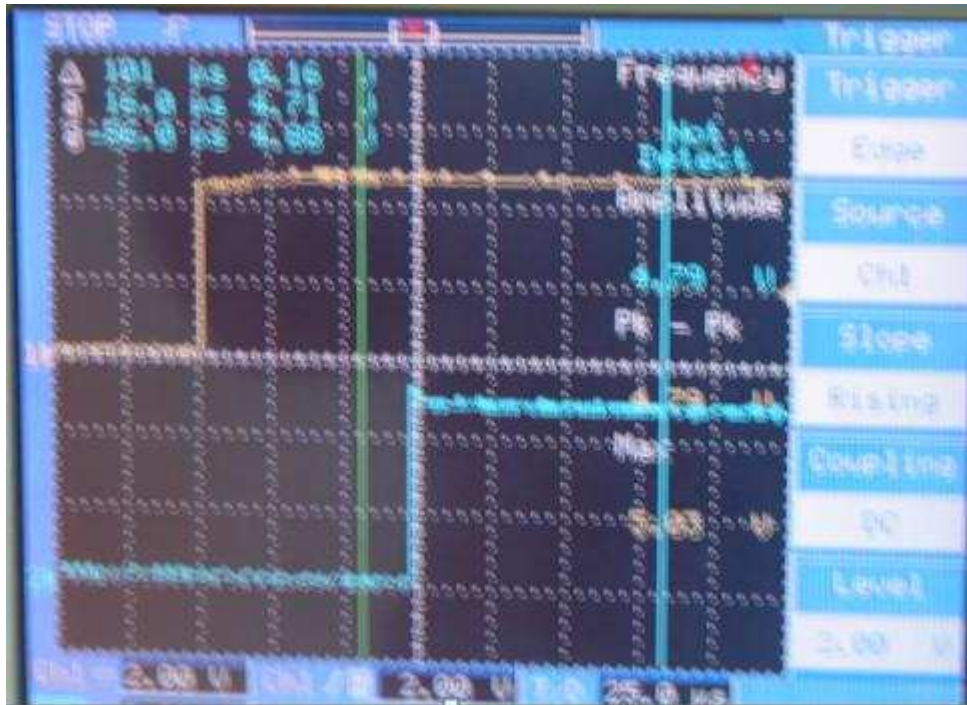


Figura 22-Imagem de medição do tempo de leitura das portas digitais

Na figura 22 pode-se ver o tempo passado entre a activação da saída e a activação do pino que indica que foram lidas as portas. O valor da saída é representada pela linha amarela, enquanto a linha azul representa o valor da porta que é activada. Analisando a imagem, pode-se ver que o tempo que medeia a activação do sinal de saída e a conclusão do processo de leitura das entradas é de $75\mu\text{s}$. Do tempo total medido, cerca de $10\mu\text{s}$ são justificados pelo tempo gasto na transmissão via SPI, transmissão de 40 bits, 8 para o envio do comando para o microcontrolador secundário, para desactivação das linhas de *chip select* do conector de expansão, mais 32, 4 bytes, para o processo de leitura do valor das 16 entradas digitais. O resto do tempo pode ser justificado, pelo tempo que o dispositivo que funciona como entradas demora a gerar a interrupção e pelo tempo gasto pelo microcontrolador com a execução do código que necessário para completar o processo de leitura.

5.6.2 Resposta a Comandos vindos do Computador

Como foi dito anteriormente, com a aplicação feita para interface com o operador, poderiam ser testados diferentes modos de funcionamento, sendo que se fosse

seleccionado o modo de funcionamento automático, seriam enviados ciclicamente comandos de escrita nas portas e de leitura do estado das entradas permitindo assim simular diferentes situações de funcionamento. As diferentes situações que são possíveis variam em dois aspectos fundamentais: no tempo de ciclo que separa o envio de conjuntos de comandos e a maneira como são manipuladas as saídas do sistema. A utilização destas diferentes situações, permitiu avaliar a resposta do sistema desenvolvido ao envio de um conjunto de comandos de escrita e leitura de forma cíclica. Assim foi avaliado o comportamento do sistema em duas situações distintas: uma, onde os comandos foram enviados de forma cíclica em intervalos de 100ms, e uma segunda, onde os comandos seriam enviados de 10 em 10ms. Em qualquer um dos casos, era apenas manipulada uma das saídas, sendo variado o seu estado de cada um dos ciclos.

Na figura 23, temos representada a variação ao longo do tempo da referida saída, para um tempo de ciclo de 100ms. Podemos ver que o seu valor varia como o pretendido, com variações cíclicas do seu estado de 100 em 100ms, sendo que este estado reflectia-se também no valor amostrado pelo computador para esta porta.

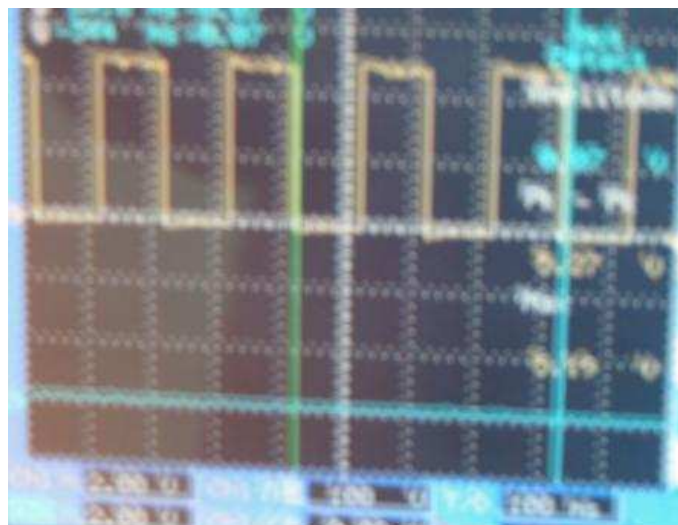


Figura 23-Variação do estado de uma saída com o envio de comandos de escrita de 100 em 100ms

Por outro lado, podemos ver na figura 24, a variação do sinal de saída com o envio de comandos de 10 em 10ms. Neste caso, podemos ver que o sinal já não reflecte a situação desejada, havendo uma variação do sinal de 20 em 20ms. Isto acontece, por que é introduzido um atrasado de 10ms, pelo processamento dos comandos recebidos

por parte do microcontrolador. Devido ao funcionamento em fila de espera dos buffers usados para a transmissão via porta série, este atrasado é cumulativo, o que faz com que o atrasado da onda produzida, em relação à forma de onda desejada aumente 10ms por cada comando enviado.



Figura 24-Variação da saída com comandos de 10 em 10ms

5.6.3 Conclusões

Analisando os resultados obtidos, pode-se concluir que o sistema desenvolvido consegue cumprir todas as funcionalidades exigidas para este tipo de sistemas, permitindo aquisição de dados e o controlo de variáveis. À que salientar, que com os testes realizados, foram testadas as situações de pior caso possível para performance do sistema, visto que eram enviados os todos os comandos de leitura e escrita, na situação que exigia mais processamento com microcontrolador e maior quantidade de dados a transmitir. Por outro lado, as situações de escrita testadas, normalmente não serão executadas nesta forma num sistema de SCADA, o que levaria a uma melhor performance do sistema. Em termos do tempo gasto, para a aquisição das entradas digitais, que o valor registado é bastante razoável permitindo a captação de variações bastante rápida do valor destas. Por último, há que referir que, a performance obtida poderá ser melhorada com a utilização de plataformas de desenvolvimento com performances ao nível do processamento mais elevada.

Capítulo 6 Conclusões

Com este trabalho, pretendia-se avaliar a viabilidade da implementação de um sistema de controlo e supervisão recorrendo ao uso do protocolo USB e de um simples computador.

Com o estudo realizado, conclui-se que o protocolo USB apresenta duas grandes desvantagens em relação às soluções actualmente utilizadas para a implementação dos sistemas de SCADA: a distância que é possível cobrir; e a estrutura que suporta o seu funcionamento, visto que apenas é possível comunicação numa esquema mestre-escravo, sendo que a sua implementação implica o uso de um computador, que funciona como mestre. Esta não seria propriamente uma desvantagem, caso a distância permitida, 5 metros, se adequasse aos requisitos deste tipo de sistemas. Por outro lado, em termos de performance esta solução permite suportar todos os requisitos de comunicação exigidos para este tipo de sistema, tanto ao nível temporal, como relativos aos tipos de tráfego exigidos, superando-os em alguns aspectos, como por exemplo, em termos de velocidade de transmissão, com taxas de transmissão máxima de 480Mbit/s, na versão 2.0, estando previsto um aumento para 4.8Gbit/s para a versão 3.0. Por outro lado, a evolução da versão *wireless* do USB permite aumentar um pouco o alcance, 10 metros, reflectindo-se este aumento numa diminuição da taxa de transmissão, 110 Mbit/s, que continua a ser suficiente para os requisitos impostos.

Apesar das desvantagens apontadas, que numa primeira análise poderiam levar ao descartar da opção USB, como suporte deste tipo de sistemas, a constante evolução da tecnologia permite pensar o contrário. Esta permite que, hoje em dia, existam computadores de tamanho bastante reduzido que possibilitam, não só o suporte do funcionamento do barramento USB, mas também, com a utilização de outras redes de comunicações, como Ethernet ou redes *Wireless*, a interconexão dos diferentes computadores utilizados, para que possam ser integrados todos os dados adquiridos, permitindo que sejam cumpridas assim as funcionalidades exigidas num sistema de SCADA. Esta evolução é sentida também ao nível dos dispositivos utilizados para monitorização e controlo, fazendo com que seja possível o desenvolvimento de plataformas de aquisição de dados cada vez mais reduzidas, permitindo assim, a implementação de soluções globais mais flexíveis.

Por outro lado, a necessidade da colocação de um computador, por cada nó onde é feita a aquisição de dados, permite que se possa descentralizar as funções de supervisão. Assim, em vez de centralizar todas estas funções no computador usado como servidor central do sistema, podem ser usadas as potencialidades dos computadores colocados junto de cada nó, para que alguns dos algoritmos de supervisão que são executados automaticamente pelo PC, possam ser executados localmente nestes computadores, permitindo melhorar assim o desempenho do sistema. A utilização destes computadores permite, não só para implementar funcionalidades de controlo, mas também a possibilidade do armazenamento de dados que depois possam ser usados para a elaboração de históricos de funcionamento, possibilitando assim a criação de uma salvaguarda para estes dados, tornando assim o sistema menos imune a falhas que possam levar à perda destes dados.

O uso de um computador, por cada célula, faz com que seja possível serem implementadas funções de visualização localmente, o que pode ser extremamente útil, em situações de avaria, visto que ao poder ser visualizado de forma simples o valor das variáveis do processo, facilmente se pode reparar a avaria. A implementação destas funcionalidades, pode ser também útil na implementação programas de melhoria de qualidade e desempenho, visto que torna possível localmente aceder-se a estatísticas de funcionamento, que tornariam a avaliação das medidas implementadas mais simples e interactiva.

Em termos económicos, esta solução releva-se bastante competitiva, visto que o valor monetário, de um computador que cumpra os requisitos mínimos para a implementação deste tipo de sistemas, é relativamente reduzido, tendo a tendência para diminuir, sendo também relativamente baixo, o custo de desenvolvimento da plataforma de aquisição de dados. Esta competitividade nota-se ainda mais comparando com as soluções utilizadas hoje em dia, recorrendo ao uso de PLCs, que são relativamente dispendiosas.

Relativamente, ao trabalho experimental realizado há a dizer, que foi conseguido implementar um sistema que cumprisse os requisitos impostos, sendo importante dizer que este é apenas introdutório, podendo ser bastante melhorado recorrendo a outro tipo de tecnologias que permitam um melhor desempenho. Um exemplo, de melhorias que poderiam ser introduzidas, seria a utilização de um microcontrolador com níveis de

desempenho mais elevados, ou recorrer-se ao uso de FPGAs, que poderiam introduzir um aumento significativo, na performance do sistema desenvolvido. Este poderia ser também melhorado, através de uma possível optimização dos protocolos de comunicação, que permitissem uma redução nos tempos de processamento apresentados. Por outro lado, as comunicações entre o computador e o microcontrolador poderiam ser melhoradas com a utilização de um dispositivo com uma interface USB embutida, permitindo assim utilizar todas as potencialidades deste protocolo.

Assim, depois de realizado este trabalho pode-se concluir, que o uso do USB como suporte de um sistema de SCADA é extremamente viável, contribuindo para a implementação de sistemas com melhor desempenho, mais flexíveis e economicamente bastante competitivos.

Referências

- [1] - Paulo José Lopes Machado Portugal, “*Avaliação da Confiança no funcionamento das redes de campo: contribuição no domínio dos sistemas industriais de controlo*”, Tese de Doutoramento em Engenharia Electrotécnica e de Computadores, FEUP, 2004
- [2] – Jean-Pierre Thomesse, “*Fieldbus Technology in Industrial Automation*”, Proceedings of the IEEE, Vol. 93, N°. 6, June 2005
- [3] –Thilo Sauter, “*The Industrial Technology Handbook*”, Capítulo 7, CRC Press 2005
- [4] –Stuart A. Boyer, “*Scada: Supervisory Control and Data Acquisition*”, 2ª Edition - ISA-1999
- [5] – Universal Serial Bus Specification Revision 2.0 – April 27, 2000
- [6] - Datasheet do dispositivo MCP23S17

<http://ww1.microchip.com/downloads/en/DeviceDoc/21952a.pdf>